



TUGAS AKHIR

(NA 1701)

ANALISA KAVITASI HYDROFOIL DENGAN *CAVITATION BUCKETS DIAGRAM*



RSPB
623.8204
Sug
2-1
2000

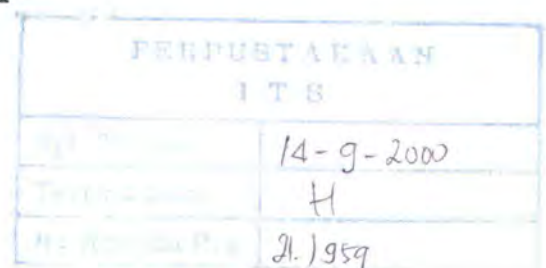
OLEH :

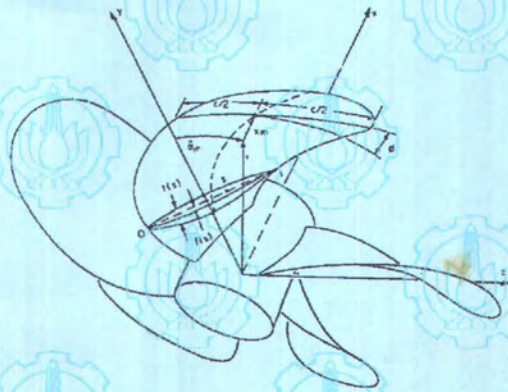
SUTAMIAJI SUGIARTO

NRP : 4193 100 045

JURUSAN TEKNIK PERKAPALAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA

2000





LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

TUGAS AKHIR

ANALISA KAVITASI HYDROFOIL DENGAN *CAVITATION BUCKETS DIAGRAM*

Menyetujui,

Jurusan Teknik Perkapalan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember

PEMBIMBING I



Ir. PETRUS ADRIANTO, MSc.

NIP. 130 325 763

LEMBAR PENGESAHAN

TUGAS AKHIR

ANALISA KAVITASI HYDROFOIL DENGAN *CAVITATION BUCKETS DIAGRAM*

Menyetujui,

Balai Penelitian & Pengkajian Hidrodinamika
BPP Teknologi Indonesia
Surabaya

PEMBIMBING II



Ir. SETYO LEKSONO, MT.

NIP. 680 002 840



ABSTRAK

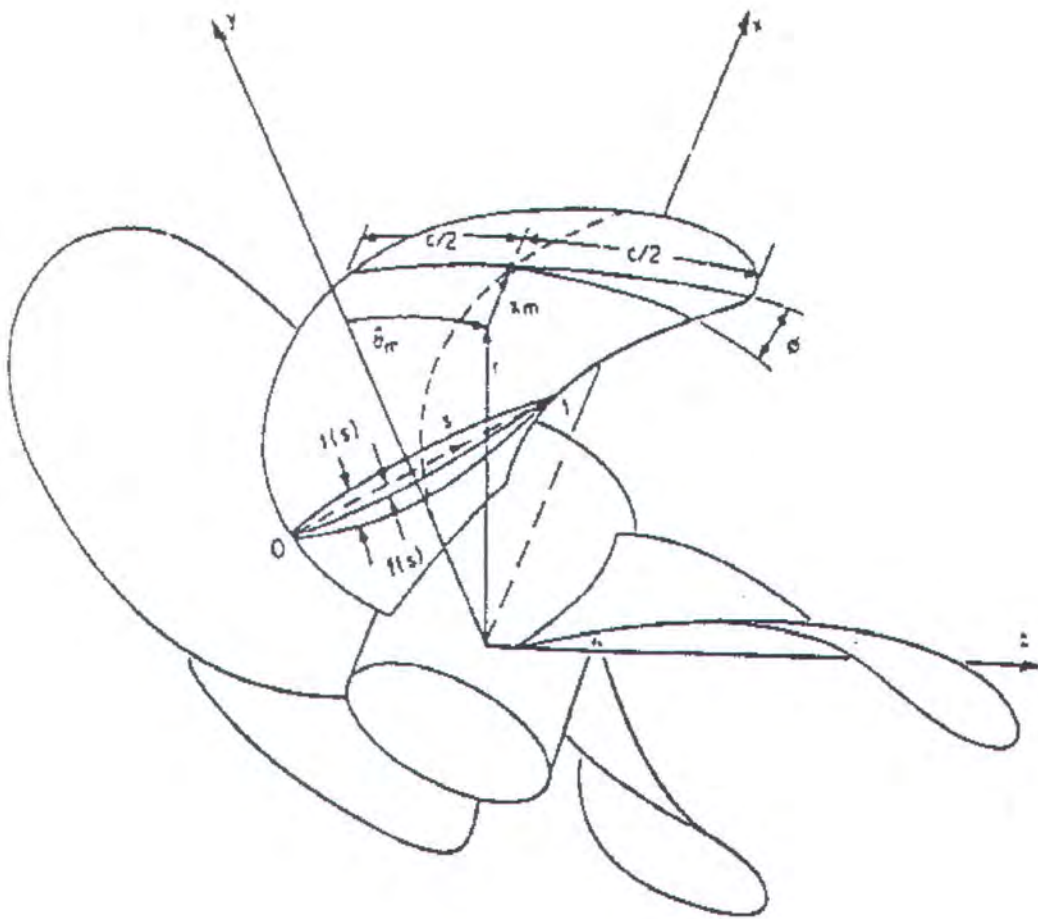


ABSTRAK

Pada disain awal suatu penampang hydrofoil dari perhitungan lift dan drag akan dianalisa lanjut mengenai kemungkinan kavitasi yang terjadi. Analisa dilakukan dengan “metode vortex panels” untuk mendapatkan grafik C_p yang menyatakan distribusi tekanan di seluruh penampang foil, pressure side dan suction side. Nilai $-C_{pmin}$ yang didapat kemudian diplotkan di dalam sebuah *cavitation bucket diagram* untuk setiap angle of attack untuk l/c tertentu. Kumpulan *cavitation buckets diagram* adalah diagram-diagram dengan l/c yang berbeda-beda.

Cavitation buckets diagram memberikan batas-batas $-C_{pmin}$ untuk suatu hydrofoil agar terhindar dari kavitasi yang merugikan. Diagram ini memberikan parameter-parameter kavitasi yaitu thickness ratio, camber ratio, dan angle of attack. Jadi dengan merubah tebal, mean line, dan angle of attack suatu foil disain mendapatkan geometri optimum dalam konteks kavitasi.

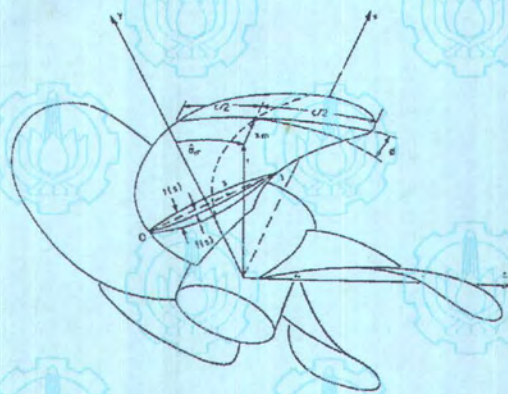




Kupersembahkan Tugas Akhir Ini

untuk Papa dan Mama

untuk my sisters Menik dan Ana



KATA PENGANTAR

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa, atas segala anugerah dan rahmat-Nya sehingga dapat menyelesaikan tugas akhir yang bertema “ANALISA KAVITASI HYDROFOIL DENGAN *CAVITATION BUCKETS DIAGRAM*”.

Ucapkan terima kasih yang sebesar-besarnya atas segala dukungan dan bantuannya, baik moril maupaun materiil, kami sampaikan kepada :

- Bpk. Ir. Jauhar Manfaat, M.Sc, Phd., Kepala Jurusan Teknik Perkapalan.
- Bpk. Ir. I K. A. Pria Utama, MSc, Phd., Sekretaris Jurusan Teknik Perkapalan.
- Bpk. Ir. Petrus Adrianto, MASc, pembimbing I.
- Bpk. Ir. Setyo Leksono, MT, pembimbing II.
- Bpk. Ir. Mahardjo Wartono, dosen wali.
- Bpk. Ir. Murdijanto, M.Eng, mantan pembimbing I.
- Bpk. Dr. Ir. Wisnu W., M.Sc dan Bpk. Ir. I G.M. Santosa.
- Bpk. Ir. Tjinto, atas segala kesempatan dan fasilitasnya di LHI.
- Bpk. Suhdiono, Bpk. Suwandi, Mas Heru, Bu Ana dan seluruh karyawan FTK-ITS.
- Seluruh staf dan karyawan Balai Penelitian dan Pengkajian Hidrodinamika BPP Teknologi Indonesia.
- Ova, Sita, Kole', dan Joni, suwun-suwuun.
- Rekan-rekan **hidrodinamika-wan** Roni Inor, Nanang AP, dan Robin S.
- Rekan-rekan se-angkatan (angkatan '93), “**viva P-33**”.

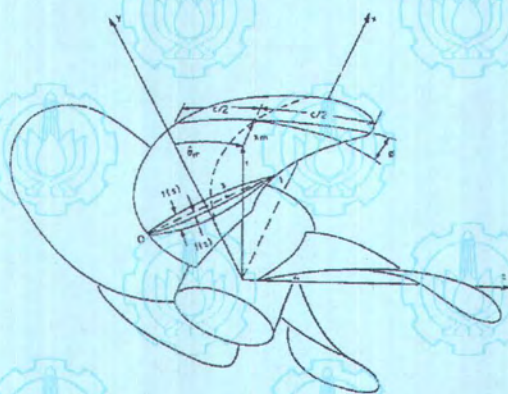
- Rekan-rekan satu kost P-3 Perumdos ITS, kostnya cowok-cowok keren, “doomo arigatoo gozaimasu”.
- Segala pihak yang tidak dapat kami sebutkan satu persatu.

Tugas akhir ini disusun sebagai salah satu prasyarat untuk meraih gelar kesarjanaan di jurusan Teknik Perkapalan FTK-ITS. Besar harapan penulis agar tugas akhir ini dapat memberikan sedikit kontribusi dalam peningkatan pemahaman tentang perencanaan baling-baling.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan tugas akhir ini, oleh karena itu saran dan kritik untuk perbaikan dan pengembangan tugas akhir ini akan diterima dengan tangan terbuka.

Surabaya, Februari 2000

P e n u l i s



DAFTAR ISI

DAFTAR ISI

LEMBAR PENGESAHAN	ii
ABSTRAK	iv
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi

BAB I PENDAHULUAN

1.1. Latar belakang	1
1.2. Permasalahan	2
1.3. Tujuan dan Manfaat	3
1.4. Batasan Masalah	3
1.5. Metodologi	4

BAB II DASAR TEORI

2.1. Fluida Ideal	6
2.2. Aliran sekitar Foil Dua Dimensi	11
2.2.1 Geometri Penampang Sayap	11
2.2.2. Sirkulasi	13
2.2.3. Aliran Uniform, Vortex, dan Silinder dengan Sirkulasi	13
2.2.4. Vortex Sheet	20
2.3. Persamaan Bernoulli	22
2.4. Teori Kutta-Joukowski	24

BAB III DISTRIBUSI TEKANAN

3.1. Metoda Panel Dua Dimensi	27
3.1.1. Aliran Potensial sekitar Hydrofoil	27
3.1.2. Perhitungan Distribusi Tekanan secara Numerik	28
3.2. Distribusi tekanan dan Awal Kavitasi	35

3.3.	Gaya Angkat	38
3.4.	Kondisi Kutta	40
3.5.	<i>Cavitation Buckets Diagram</i>	41

BAB IV ANALISA HASIL RUNNING PROGRAM

4.1.	Algoritma Program	46
4.2.	Perubahan Thickness Ratio dan Camber Ratio	51
4.3.	Pengaruh Perubahan Thickness Ratio terhadap Angle of Attack dan Koefisien Gaya Angkat	54
4.4.	Pengaruh Perubahan Camber Ratio terhadap Angle of Attack dan Koefisien Gaya Angkat	57
4.5.	Batas Kavitasi	60
4.6.	Panjang Kavitasi dan Jenis Kavitasi yang Terjadi	61
4.7.	Geometri Foil yang Optimum	64

BAB V KESIMPULAN DAN SARAN

5.1.	Kesimpulan	66
5.2.	Saran	67

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

TABEL	KETERANGAN
4.1	Data $-C_{pmin}$ setiap angle of attack karena perubahan t/c
4.2	Data $-C_{pmin}$ setiap angle of attack karena perubahan f/c

DAFTAR GAMBAR

Gb. 2.1	Sayap tiga dimensi	11
Gb. 2.2	Geometri penampang sayap	12
Gb. 2.3	Sirkulasi dalam bidang	13
Gb. 2.4	Aliran uniform	14
Gb. 2.5	Koordinat kutub	14
Gb. 2.6	Aliran vortex	15
Gb. 2.7	Aliran rektilinier tanpa vortex melewati sebuah silinder	17
Gb. 2.8	Aliran rektilinier dan vortex nonrotasi sekitar sebuah silinder	18
Gb. 2.9	Vortex sheet	20
Gb. 2.10	Lompatan kecepatan tangensial yang melintasi vortex sheet	21
Gb. 2.11	Sirkulasi pada penampang sayap	25
Gb. 3.1	Hydrofoil dalam aliran fluida	28
Gb. 3.2	Penempatan vortex panel dengan kekuatan vortex linier	31
Gb. 3.3	Penentuan titik batas dan titik kontrol vortex panel	34
Gb. 3.4	Distribusi vortex tiap titik kontrol panel	35
Gb. 3.5	Kecepatan fluida yang terganggu oleh foil	36
Gb. 3.6	Aliran sekitar penampang dengan sirkulasi nol	41
Gb. 3.7	Koefisien tekanan lawan angle of attack	43
Gb. 3.8	Koefisien tekanan lawan koefisien gaya angkat	44
Gb. 3.9	Geometri dan koefisien gaya angkat pada angka kavitasi	45
Gb. 4.1	Metode kombinasi mean line dan basic thickness form	52
Gb. 4.2	Cavitation buckets diagram dengan parameter thickness ratio	56
Gb. 4.3	Sebuah buckets diagram untuk camber ratio yang berbeda	58
Gb. 4.4	Cavitation buckets diagram untuk camber ratio 0,06	59
Gb. 4.5	Cavitation buckets diagram untuk camber ratio 0,04	59
Gb. 4.6	Cavitation buckets diagram untuk camber ratio 0,02	59
Gb. 4.7	Cavitation buckets diagram dan angka kavitasi	60
Gb. 4.8	Sebuah bucket diagram yang dibentuk oleh $-C_{pmin}$ tiap panel	62
Gb.4.9	Jenis kavitasi yang terjadi	64



BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1. Latar Belakang

Disain foil mempunyai beberapa parameter seperti lift, drag, kavitasi dan kekuatan. Lift ini akan memberikan gaya angkat pada foil dan thrust pada baling-baling sehingga foil banyak digunakan pada aplikasi teknik. Drag akan memberikan tahanan pada foil dan torque pada baling-baling. Kekuatan akan memperhitungkan kemampuan dari sayap atau baling-baling untuk menahan beban kerjanya. Sedangkan kavitasi dapat timbul pada bidang permukaan angkat (lifting surface) foil dan ini hanya terjadi pada hydrofoil yaitu foil yang bekerja di dalam fluida incompressible (air).

Kavitasi merupakan fenomena alam apabila air yang mengalir melalui daerah bertekanan rendah mencapai vapour pressure. Kavitasi ini memang tidak diinginkan oleh karena itu disain foil yang sudah ada hasil perhitungan lift dan drag-nya

dianalisa apakah kavitasi yang terjadi berada dalam batas-batas yang membahayakan atau tidak.

Pada sisi hidrodinamika, efek kavitasi akan mengakibatkan defisiensi kinerja permukaan angkat, getaran, dan noise yang besar. Kemudian efek yang terjadi akan berpengaruh juga pada material daun baling-baling karena gelembung-gelembung yang terjadi terbawa aliran fluida dan akan pecah bila memasuki daerah aliran bertekanan lebih besar. Pecahnya gelembung ini akan menghantam permukaan daun baling-baling sehingga material akan tererosi, hal ini akan mengurangi kekuatan baling-baling tersebut.

Salah satu disain foil untuk masalah kavitasi adalah dengan memodelkan secara numerik dengan menggunakan model matematis dua dimensi yang kemudian dianalisa pada *cavitation buckets diagram*. Pemakaian metode numerik memiliki kelemahan diantaranya yaitu banyaknya asumsi yang digunakan, akan tetapi dipihak lain analisa dapat dilakukan dengan cepat dan tidak terikat oleh bentuk model.

1.2. Permasalahan

Pada tugas akhir ini permasalahan yang akan dicoba untuk diselesaikan adalah apakah *cavitation buckets diagram* dapat menganalisa kavitasi dengan menggunakan metode panel dua dimensi. Dengan demikian akan diprediksi geometri hydrofoil dengan *angle of attack*, *thickness ratio*, dan *camber ratio*.

1.3. Tujuan dan Manfaat

Seperti telah kita ketahui, kavitasi yang tidak terkendali akan merubah bentuk aliran disekitar foil sehingga akan berpengaruh terhadap kinerja foil dan juga baling-baling. Untuk itu hydrofoil perlu didisain untuk menghindari kavitasi yang sangat merugikan. Hasil analisa kavitasi ini akan menghasilkan geometri optimum untuk masing-masing camber ratio dalam konteks *cavitation buckets diagram*.

Dengan dipilihnya metode numerik ini, maka dapat digunakan sebagai sub routine dari program komputer untuk mendisain baling-baling atau sayap kapal hydrofoil secara utuh.

1.4. Batasan Masalah

Pada pembahasan tugas akhir ini ada beberapa batasan yang akan digunakan sehingga proses perhitungan menjadi lebih sederhana, yaitu :

- Aliran akan dihitung pada ruang dua dimensi.
- Foil diam di dalam fluida yang mengalir.
- Untuk analisa karakteristik aliran di sekitar foil digunakan asumsi fluida ideal sehingga mempunyai sifat :
 - * Tidak dapat dimampatkan (incompressible), yaitu massa jenis fluida tersebut konstan.
 - * Tidak viscous (invisid), yaitu tidak memiliki kekentalan atau gesekan di dalam.

- * Steady (tidak bergantung waktu), jadi kecepatan zat cair pada suatu titik hanya dari posisinya (x,y) dan tidak bergantung waktu.
- * Tidak berotasi (irrotational), yaitu bilamana sikulasinya nol.

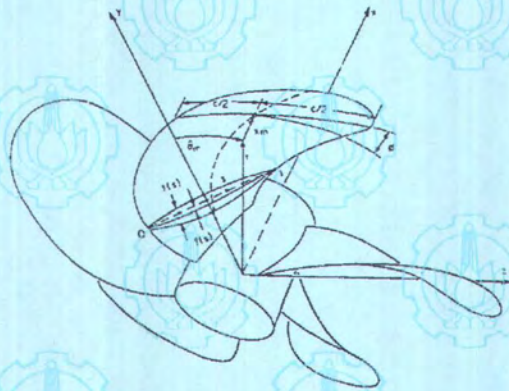
1.5. Metodologi

Pada umumnya penulisan tugas akhir untuk mahasiswa S_1 berupa studi terhadap suatu masalah tertentu, untuk itu kita mengacu pada acuan yang sudah pasti dipandang benar. Acuan tersebut didapat dari berbagai referensi atau organisasi yang sudah kita kenal eksistensinya dalam permasalahan yang jadi perhatian kita. Sehingga dengan studi bisa diperoleh pemahaman yang lebih mendalam terhadap permasalahan yang kita pelajari.

Pada penulisan tugas akhir ini, penulis melakukan studi terhadap permasalahan seperti tersebut di dalam judul, dengan langkah-langkah sebagai berikut :

- Analisa ini membutuhkan bentuk foil yang sudah jadi dalam suatu preliminari design yang sudah melalui tahap perhitungan lift.
- Foil kasus diambil dari seri NACA empat-digit.
- Menghitung distribusi tekanan disekitar foil dua dimensi dari foil kasus dengan menggunakan metode vortex panel untuk mendapatkan nilai distribusi C_p dan C_L sepanjang chord length.

- Menentukan titik-titik di permukaan foil yang mempunyai tekanan minimum untuk setiap angle of attack dengan jangkauan -5 derajat sampai 6 derajat.
- Variasi tekanan minimum ($-C_p \text{ min}$) dan angle of attack diplotkan untuk mendapatkan sebuah *cavitation bucket diagram*.
- Perhitungan diulang untuk thickness ratio t/c yang berbeda sehingga didapatkan satu set *cavitation buckets diagram*.
- Perhitungan diulang untuk camber ratio f/c yang berbeda sehingga didapatkan satu seri *cavitation buckets diagram*.
- Beberapa analisa yang dilakukan adalah pengaruh thickness ratio terhadap angle dan koefisien gaya angkat, pengaruh camber ratio terhadap angle of attack dan koefisien gaya angkat, angka kavitasi, dan jenis kavitasi yang terjadi.
- Menampilkan geometri foil yang optimum sesuai kriteria kavitasi..



BAB II

DASAR TEORI

BAB II

DASAR TEORI

2.1. Fluida Ideal

Untuk analisa karakteristik aliran sekitar penampang sayap, digunakan asumsi aliran fluida mempunyai sifat : incompressible, non viscous, steady, irrotational dan aliran dua dimensi.

Hukum fisika dasar yang harus dipatuhi oleh partikel fluida yang bergerak, yaitu tiga hukum kekekalan. Secara umum persamaan pergerakan fluida yang bersifat viscous, mampat dan tak mampat diturunkan dari persamaan Navier-Stokes yang dapat dinyatakan dalam bentuk vektor dari 3 persamaan berikut :

1. Persamaan Kontinuitas (Hukum Kekekalan Massa)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \bar{v} = 0$$

dimana : ρ = densiti fluida

\bar{v} = kecepatan fluida

2. Persamaan Momentum (Hukum Kekekalan Momentum)

$$\rho \frac{D\bar{v}}{Dt} = \rho \bar{f} - \nabla P - \frac{2}{3} [\mu (\nabla \cdot \bar{v})] + [\nabla (2\mu \cdot \nabla)] \bar{v} + \nabla x [\mu \nabla x \bar{v}]$$

dimana :

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z}$$

\bar{f} = gaya badan (body force)

P = tekanan

μ = viskositas

3. Persamaan Energi (Hukum Kekekalan Energi)

$$\rho \frac{dQ}{dt} + \phi_{viscous} = \rho \frac{Dh}{Dt} - \frac{DP}{Dt}$$

dimana : Q = energi panas

ϕ = potensial kecepatan

h = jarak perpindahan

Kemudian dari asumsi :

Incompressible, berarti ρ adalah konstan atau $\frac{\partial \rho}{\partial t} = 0$, sehingga persamaan

kontinuitas dapat disederhanakan menjadi $\nabla \cdot \bar{v} = 0$ atau

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \dots\dots\dots(2.1)$$

Inviscid, berarti mempunyai viskositas yang sangat kecil dan dapat diabaikan ($\mu = 0$) dan gaya badan (\bar{f}) berharga sangat kecil, sehingga persamaan momentum dapat disederhanakan menjadi :

$$\rho \frac{Dv}{Dt} = -\nabla P \quad \text{atau} \quad \frac{Dv}{Dt} = -\frac{1}{\rho} \nabla P$$

sehingga

$$\text{untuk komponen x : } \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = -\frac{1}{\rho} \nabla P \quad \dots\dots\dots(2.2a)$$

$$\text{untuk komponen y : } \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} = -\frac{1}{\rho} \nabla P \quad \dots\dots\dots(2.2b)$$

$$\text{untuk komponen z : } \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho} \nabla P \quad \dots\dots\dots(2.2c)$$

sedangkan persamaan energi dapat diabaikan karena fluida dalam kondisi incompressible dan steady.

Untuk fluida yang tak mengalami rotasi, vortisitas (ω) adalah

$$\bar{\omega} = \nabla \times \bar{v} \quad \text{atau} \quad \bar{\omega} = \begin{vmatrix} x & y & z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ u & v & w \end{vmatrix} = \omega_x \cdot x + \omega_y \cdot y + \omega_z \cdot z \quad \dots\dots\dots(2.3)$$

dimana :

$$\omega_x = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} ; \quad \omega_y = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} ; \quad \omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

karena irrotational, maka $\bar{\omega} = 0$ dan dibatasi dalam aliran dua dimensi sehingga persamaan (2.3) menjadi :

$$\bar{\omega} = \omega_z = \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) z = 0 \quad \text{atau} \quad \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = 0 \quad \dots\dots\dots(2.4)$$

Dari persamaan (2.1) untuk aliran dua dimensi dapat disederhanakan menjadi :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad \dots\dots\dots(2.5)$$

Dari persamaan (2.2) untuk aliran dua dimensi dapat disederhanakan menjadi :

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial x}$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial y}$$

Bila keduanya diturunkan parsial terhadap x dan y , maka akan menjadi :

$$\frac{\partial^2 u}{\partial t \partial y} + u \frac{\partial^2 u}{\partial x \partial y} + \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} + v \frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial^2 P}{\partial x \partial y}$$

$$\frac{\partial^2 v}{\partial t \partial x} + u \frac{\partial^2 v}{\partial x^2} + \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + v \frac{\partial^2 v}{\partial x \partial y} + \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial^2 P}{\partial x \partial y}$$

Selisih dari dua persamaan di atas apabila disubstitusikan ke persamaan (2.4)

mempunyai nilai nol, adalah sebagai berikut :

$$\frac{\partial}{\partial t} \left[\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right] + u \frac{\partial}{\partial x} \left[\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right] + \frac{\partial u}{\partial x} \left[\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right] + v \frac{\partial}{\partial y} \left[\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right] + \frac{\partial v}{\partial y} \left[\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right] = 0$$

Sehingga persamaan akhir atur pergerakan fluida (persamaan Navier-Stokes)

adalah persamaan (2.4) dan (2.5). Kemudian dari persamaan (2.4) didapatkan :

$$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = 0$$

yang mempunyai penyelesaian

$$\partial v = \int \frac{\partial u}{\partial y} \partial x \quad \text{atau} \quad v = \frac{\partial \phi}{\partial y} \dots\dots\dots(2.6a)$$

$$\partial u = \int \frac{\partial v}{\partial x} \partial y \quad \text{atau} \quad u = \frac{\partial \phi}{\partial x} \dots\dots\dots(2.6b)$$

Penyelesaian dari persamaan (2.4) di atas dikenal dengan nama potential function dimana persamaan tersebut juga harus memenuhi persamaan (2.5), sehingga substitusi persamaan (2.6) ke persamaan (2.5) akan memberikan

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \nabla^2 \phi = 0$$

yaitu persamaan Laplace dalam variabel ϕ .

Kemudian dengan cara yang sama seperti pada persamaan (2.4), maka dari persamaan (2.5) didapatkan fungsi aliran yang mempunyai sifat :

$$u = \frac{\partial \psi}{\partial y} \quad \text{dan} \quad v = -\frac{\partial \psi}{\partial x} \dots\dots\dots(2.7)$$

dimana fungsi tersebut juga harus memenuhi persamaan (2.5), sehingga hasil substitusi persamaan (2.7) ke persamaan (2.5) adalah :

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \nabla^2 \psi = 0$$

yaitu persamaan Laplace dalam variabel ψ .

Kondisi inilah yang dipakai sebagai asumsi aliran fluida di sekitar penampang sayap, dimana hubungan komponen kecepatannya adalah sebagai berikut :

$$u = \frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} \quad \text{dan} \quad v = \frac{\partial \phi}{\partial y} = -\frac{\partial \psi}{\partial x} \dots\dots\dots(2.8)$$

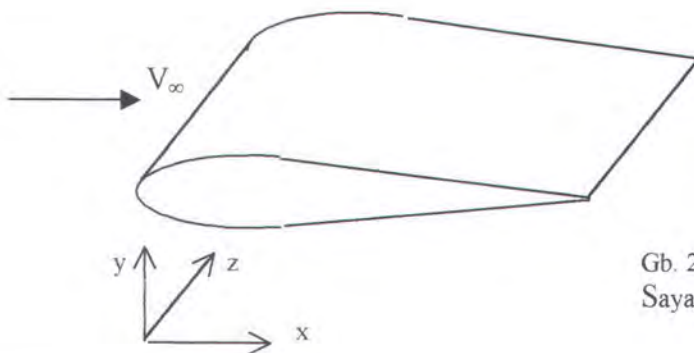


yang dikenal sebagai kondisi Cauchy-Riemann. Ini berarti bahwa untuk kurva ϕ dan kurva ψ akan saling tegak lurus dan ini merupakan hal penting untuk menyelesaikan berbagai masalah dalam bidang mekanika fluida.

2.2. Aliran sekitar Foil Dua Dimensi

2.2.1. Geometri dari Penampang Sayap

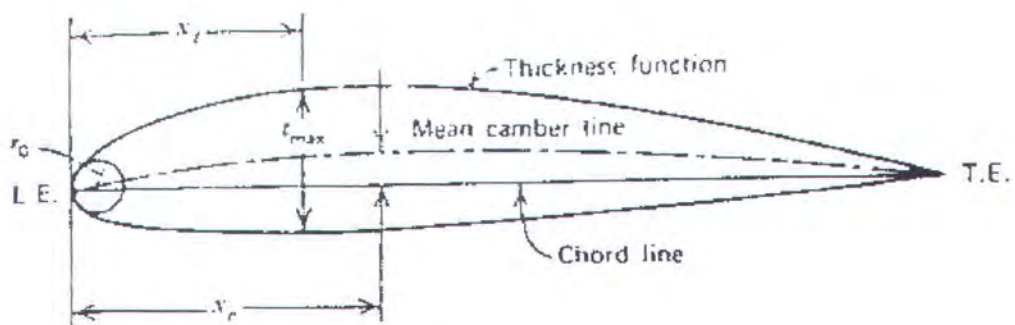
Suatu bentuk sayap yang memanjang arah $-\infty$ dan ∞ sumbu z dalam gb. 2.1 akan ditinjau sebagai penampang dua dimensi yaitu bidang x - y . Kecepatan aliran bebas V_∞ diukur pada jarak yang jauh dari penampang sayap sejajar dengan bidang x - y .



Gb. 2.1
Sayap tiga dimensi

Bagian-bagian penting dari sayap dua dimensi seperti yang tertera dalam gb. 2.2 di bawah ini adalah sebagai berikut :

- Chord line (x-x), yaitu garis lurus referensi yang digambar dari ujung depan sampai ujung belakang foil.
- Camber line , yaitu garis yang membagi foil menjadi dua bagian yang sama tebal antara permukaan atas dan bawah.
- Chord length (c), yaitu panjang penampang foil dari ujung depan ke belakang diukurkan pada chord line.
- Maximum thickness, diukur tegak lurus pada camber line, sering diekspresi yang menunjukkan perbandingan antara tebal dan chord length (t/c).
- Camber (d), yaitu ketinggian camber line maksimum dari chord line, sering dinyatakan dengan perbandingan antara nilai maksimum dan chord length (d/c).
- Angle of attack (α), yaitu sudut yang dibentuk oleh chord line dan vektor kecepatan aliran bebas.



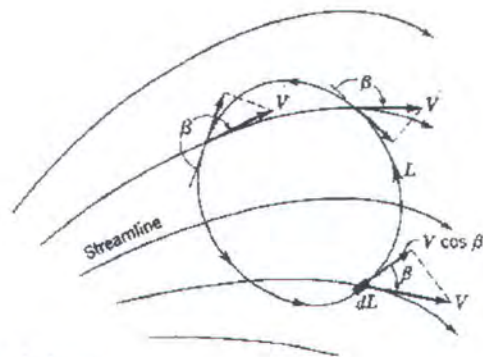
Gb. 2.2 Geometri penampang sayap

2.2.2. Sirkulasi

Sirkulasi didefinisikan sebagai integral garis melalui kurva tertutup pada arah mengikuti jarum jam dari komponen singgung lintasan kali panjang busurnya.

Besarnya sirkulasi Γ pada kurva tertutup akan selalu konstan di setiap titik.

$$\Gamma = \oint_L \mathbf{v} \cdot d\mathbf{L} = \oint_L v \cos \beta dL \quad \dots\dots\dots(2.9)$$



Gb. 2.3 Sirkulasi dalam bidang

2.2.3. Aliran Uniform, Aliran Vortex, dan Silinder dengan Sirkulasi

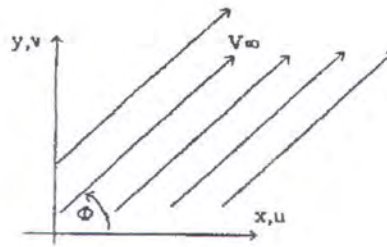
Beberapa aliran yang penting dapat dibangun dari tiga jenis penyelesaian elementer, yaitu aliran uniform, source dan sink, dan vortex. Alur aliran dapat dibangun dengan cara superposisi antara lain uniform dan vortex atau antara uniform dan source-sink. Hal ini dapat terbentuk karena potential function ketiga aliran itu adalah linier jika

$$\nabla^2 \phi_1 = 0 \quad \text{dan} \quad \nabla^2 \phi_2 = 0$$

maka $\nabla^2 (\phi_1 + \phi_2) = 0$

Aliran Uniform

Suatu aliran seragam dengan kecepatan tetap v_∞ dan arahnya membentuk sudut α terhadap sumbu x dapat dilihat pada gambar berikut :



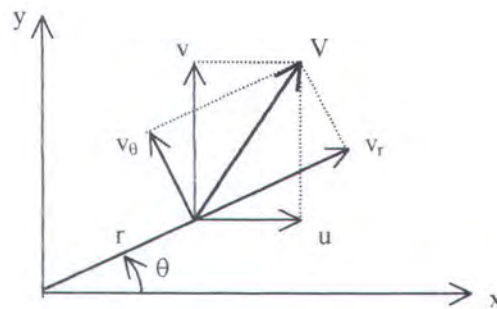
Gb. 2.4 Aliran uniform

$$u = v_\infty \cdot \cos \alpha = \frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} \dots \dots \dots (2.10)$$

$$v = v_\infty \cdot \sin \alpha = \frac{\partial \phi}{\partial y} = -\frac{\partial \psi}{\partial x} \dots \dots \dots (2.11)$$

Aliran Vortex

Untuk mendefinisikan aliran source-sink secara matematis digunakan sistem koordinat polar. Sistem koordinat ini memudahkan pendefinisian aliran ini karena komponen kecepatan diproyeksikan dalam dua parameter yaitu jari-jari dan sudut θ seperti pada gambar.



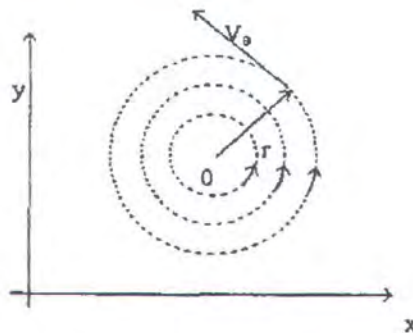
Gb. 2.5 Koordinat kutub

Persamaan stream function dan potential function untuk koordinat kutub adalah :

$$v_r = \frac{1}{r} \frac{\partial \psi}{\partial \theta} = \frac{\partial \phi}{\partial r} \dots\dots\dots(2.12a)$$

$$v_\theta = -\frac{\partial \psi}{\partial r} = \frac{1}{r} \frac{\partial \phi}{\partial \theta} \dots\dots\dots(2.12b)$$

Suatu vortex adalah aliran-aliran yang streamline-nya adalah lingkaran konsentris seperti pada gambar di bawah ini



Gb. 2.6 Aliran vortex

Pola ini disebut free vortex yaitu berupa aliran yang berputar melingkar dengan satu komponen kecepatan singgung yang mengecil sesuai $1/r$, kecepatan ini disebut kecepatan tangensial yang dinotasikan v_θ dengan kekuatan sebesar γ , sedangkan komponen kecepatan radial yang dinotasikan v_r adalah nol.

$$v_r = \frac{\partial \phi}{\partial r} = \frac{1}{r} \frac{\partial \psi}{\partial \theta} = 0$$

$$v_\theta = V = \frac{K}{2\pi r} = \frac{1}{r} \frac{\partial \phi}{\partial \theta} = -\frac{\partial \psi}{\partial r}$$

sehingga menghasilkan

$$\phi = \frac{K}{2\pi} \theta \quad \dots\dots\dots(2.14a)$$

$$\psi = -\int \frac{K}{2\pi r} dr = -\frac{K}{2\pi} \ln r \quad \dots\dots\dots(2.14b)$$

Konstanta K dapat dievaluasi dengan mengambil persamaan sirkulasi

$$\Gamma = \int_s V_s dS$$

Untuk kasus khusus integral garis sepanjang kurva tertutup berupa lingkaran yang mengelilingi suatu titik yang merupakan pusat sebuah vortex, vektor kecepatannya v_θ selalu konstan dan bersinggungan dengan lintasan integrasi, dan $dS = r.d\theta$. Jadi

$$\begin{aligned} \Gamma &= v_\theta \oint dS = v_\theta \int_0^{2\pi} r.d\theta \\ \Gamma &= 2\pi v_\theta r \quad \dots\dots\dots(2.15) \end{aligned}$$

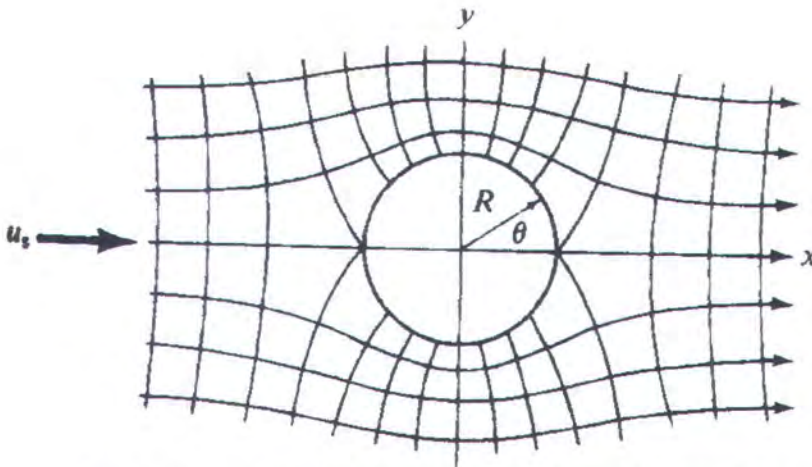
dan karena $v_\theta = \frac{K}{2\pi r} \quad \Gamma = K \quad \dots\dots\dots(2.16)$

Silinder dengan Sirkulasi

Aliran di seputar sebuah silinder bundar dengan sirkulasi didapatkan dengan menambahkan sebuah aliran vortex pada aliran rektilinear yang terhalang oleh silinder (vortex searah jarum jam dalam gb. 2.8). Kekuatan vortex tersebut adalah Γ .

$$\begin{aligned} \phi &= -u_s \left(x + \frac{R^2 x}{x^2 + y^2} \right) + \frac{\Gamma}{2\pi} \tan^{-1} \frac{y}{x} \\ \phi &= -u_s \left(r + \frac{R^2}{r} \right) \cos \theta + \frac{\Gamma}{2\pi} \theta \quad \dots\dots\dots(2.17) \end{aligned}$$

$$\begin{aligned}\psi &= -u_s \left(y - \frac{R^2 y}{x^2 + y^2} \right) - \frac{\Gamma}{4\pi} \ln(x^2 + y^2) \\ \psi &= -u_s \left(r - \frac{R^2}{r} \right) \sin \theta - \frac{\Gamma}{2\pi} \ln r\end{aligned} \quad \dots\dots\dots(2.18)$$



Gb. 2.7 Aliran rektilinier tanpa vortex melewati sebuah silinder

Bila seputar silinder tidak ada sirkulasi, titik-titik stagnasi (tempat kecepatan sama dengan nol) terletak pada $\theta = 0$ dan $\theta = \pi$ (gb. 2.7). Ketika sirkulasi terjadi dan bertambah, titik-titik stagnasi bergerak saling mendekat di sepanjang permukaan silinder sampai akhirnya berhimpit pada $\theta = -\pi/2$. Jadi sirkulasi diperbesar lagi, titik-titik stagnasi akan berpindah dari permukaan silinder dan fluida akan berotasi secara sempurna seputar silinder. Kecepatan tangensial di sepanjang permukaan silinder adalah

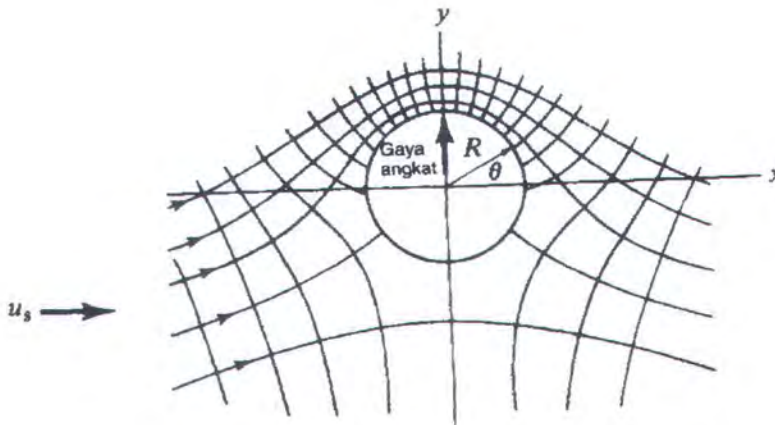
$$v_\theta = - \left(2u_s \sin \theta + \frac{\Gamma}{2\pi R} \right) \quad \dots\dots\dots(2.19)$$

(ada tanda negatif karena aliran dalam gb. 2.8 searah jarum jam), sehingga lokasi titik-titik stagnasi $v_\theta = 0$ dapat dikaitkan dengan sirkulasi melalui persamaan

$$\Gamma = -4\pi u_s R \sin \theta \quad \dots\dots\dots(2.20)$$

Sebuah titik-titik stagnasi rangkap (pada $\theta = -\pi/2$) terjadi bila kecepatan akibat sirkulasi itu dua kali u_s , dari persamaan

$$v_{sirk} = \frac{\Gamma}{2\pi R} = -2u_s \sin \theta \Big|_{v_\theta=0} = 2u_s \Big|_{v_\theta=0} \quad \dots\dots\dots(2.21)$$



Gb. 2.8 Aliran rektilinier dan vortex nonrotasi di sekitar sebuah silinder

Dalam semua situasi sebelum terjadinya pertemuan, jika kecepatan arus bebas u_s diketahui, maka kecepatan di titik lain mana pun dapat diperoleh dari potential velocity function, stream function atau pengukuran selang (spacing) ekuipotensial pada jaring aliran (*flow net*). Jika tekanan p_s diketahui di tempat kecepatan u_s juga diketahui, tekanan pada titik-titik lain dapat diketahui dari persamaan Bernoulli

$$\rho \frac{u_s^2}{2} + p_s = \rho \frac{v_1^2}{2} + p_1$$

asalkan efek gravitasi diabaikan.

Drag atau gaya hambat didefinisikan sebagai gaya resultan fluida pada permukaan sebuah benda yang arahnya sejajar dengan kecepatan arus bebas pendekatan u_s . *Lift* atau gaya angkat adalah komponen gaya yang tegak lurus terhadap arus bebas. Dalam sebuah fluida ideal, yang bekerja hanyalah gaya-gaya normal terhadap tekanan.

Untuk aliran yang melewati sebuah silinder lingkaran tanpa sirkulasi, variasi tekanan seputar permukaan dihitung dengan

$$p - p_s = \frac{\rho}{2} \left[u_s^2 - (2u_s \sin \theta)^2 \right] \dots\dots\dots(2.22)$$

Karena sirkulasi nol maka

$$drag = - \int_0^{2\pi} (p - p_s) \cos \theta (R d\theta) = 0 \dots\dots\dots(2.23a)$$

$$lift = - \int_0^{2\pi} (p - p_s) \sin \theta (R d\theta) = 0 \dots\dots\dots(2.23b)$$

Untuk aliran melalui sebuah silinder lingkaran dengan sirkulasi mempunyai variasi tekanan seputar permukaan

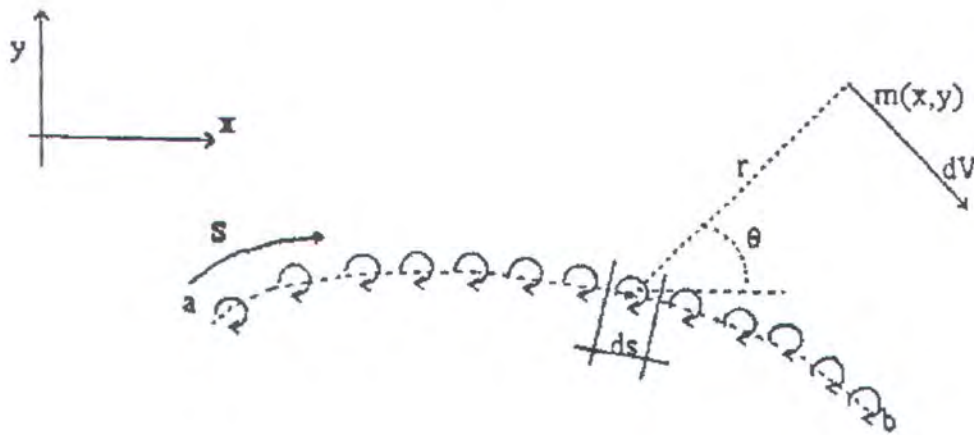
$$p - p_s = \frac{\rho}{2} \left[u_s^2 - \left(2u_s \sin \theta + \frac{\Gamma}{2\pi R} \right)^2 \right] \dots\dots\dots(2.24)$$

dan *viscous drag* sama dengan nol. Gaya angkat untuk satu satuan panjang silinder adalah

$$L = \rho \cdot u_s \cdot \Gamma \dots\dots\dots(2.25)$$

2.2.4. Vortex Sheet

Sebuah garis lurus yang melalui titik O seperti pada gb. 2.6 di atas dengan panjang yang tak berhingga dinamakan *straight vortex filament* dengan kekuatan Γ . Jika *straight vortex filament* ini jumlahnya tak berhingga dan berdampingan, maka akan terbentuk *vortex sheet* seperti pada gambar



Gb. 2.9 Vortex sheet

dimana ds adalah jarak yang diukur sepanjang S , maka kekuatan dari bagian yang sangat kecil (ds) dari vortex sheet adalah γ_s . Selanjutnya titik m berada dalam aliran yang diletakkan sejauh r dari ds , koordinat cartesian m adalah (x,y) . Bagian kecil dari vortex sheet dengan kekuatan $\gamma_s ds$ menyebabkan kecepatan yang sangat kecil dV pada titik m .

$$dV = \frac{\gamma_s ds}{2\pi r} \dots\dots\dots (2.26)$$

dengan arah tegak lurus terhadap r .

Kecepatan titik m memiliki arah yang berubah-ubah sepanjang vortex sheet dari a ke b . Oleh karena itu peningkatan potensial kecepatan $d\phi$ yang disebabkan oleh elemen vortex $\gamma_s ds$ adalah

$$d\phi = -\frac{\gamma_s}{2\pi r} \theta \dots\dots\dots(2.27)$$

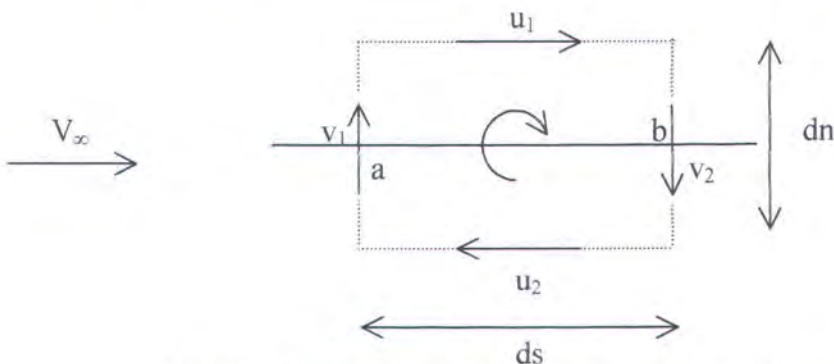
Sehingga besarnya potensial kecepatan di titik m merupakan integral dari persamaan (2.17), yaitu :

$$\phi = \int_a^b -\frac{\gamma_s}{2\pi r} \theta ds \dots\dots\dots(2.28)$$

sedangkan besarnya sirkulasi di sekitar titik vortex sama dengan kekuatan vortex itu sendiri. Dengan demikian sirkulasi yang terjadi di sekitar vortex sheet adalah merupakan jumlah dari seluruh kekuatan elemen vortex, yaitu :

$$\Gamma = \int \gamma_s ds \dots\dots\dots(2.29)$$

Karena vortex sheet mengalami perubahan diskontinuitas dalam komponen kecepatan tangensial yang melewati sheet, maka hubungan perubahan kecepatan tangensial saat melewati vortex sheet dengan kekuatan vortex adalah sebagai berikut



Gb. 2.10 Lompatan kecepatan tangensial yang melintasi vortex sheet

Anggaplah persegi panjang yang terputus-putus yang melintasi bagian dari sheet dengan panjang ds . Kecepatan tangensial bagian atas dan bawah dari lintasan ini adalah u_1 dan u_2 , sedangkan komponen samping kiri dan kanan adalah v_1 dan v_2 . Perbedaan lintasan bagian atas dan bawah adalah dn . Dari definisi sirkulasi, besarnya sirkulasi di sekitar lintasan itu adalah :

$$\Gamma = \int_a^b (u_1 + u_2)ds + (v_1 + v_2)dn \dots\dots\dots(2.30)$$

Sehingga persamaan (2.19) dan (2.20) di dapat hubungan sebagai berikut :

$$\begin{aligned} \int_a^b \gamma_s \cdot ds &= \int_a^b (u_1 + u_2)ds + (v_1 + v_2)dn \\ \gamma_s \cdot ds &= (u_1 + u_2)ds + (v_1 + v_2)dn \dots\dots\dots(2.31) \end{aligned}$$

Bagian atas dan bawah lintasan ini akan mendekati vortex sheet bila $dn \rightarrow 0$. Dengan harga limit ini, maka harga u_1 dan u_2 menjadi komponen tangensial dari vortex sheet dan seluruh bagian dan bawah dari persamaan (2.21) menjadi :

$$\begin{aligned} \gamma_s \cdot ds &= (u_1 + u_2)ds \\ \gamma_s &= (u_1 + u_2) \dots\dots\dots(2.32) \end{aligned}$$

Persamaan (2.22) ini menyatakan bahwa lokal kecepatan tangensial sama dengan kekuatan vortex sheet lokal.

2.3. Persamaan Bernoulli

Fluida ideal yang mengabaikan viskositas dengan menerapkan hukum-hukum Newton akan menghasilkan persamaan Euler untuk aliran fluida, yaitu :

$$\rho \frac{du}{dt} = -\frac{\partial P^*}{\partial x} \dots\dots\dots(2.33a)$$

$$\rho \frac{dv}{dt} = -\frac{\partial P^*}{\partial y} \dots\dots\dots(2.33b)$$

$$\rho \frac{dw}{dt} = -\frac{\partial P^*}{\partial z} \dots\dots\dots(2.33c)$$

Untuk mempermudah perhitungan, hal ini diasumsikan bahwa u, v, dan w bervariasi hanya terhadap ruang, yaitu $\frac{\partial u}{\partial t}; \frac{\partial v}{\partial t}; \frac{\partial w}{\partial t}$ adalah nol.

Jika persamaan (2.23) dikalikan dengan turunan dx, dy, dan dz, yang bersesuaian dan kemudian dihitung, hasilnya adalah :

$$\rho \left(\frac{du}{dt} dx + \frac{dv}{dt} dy + \frac{dw}{dt} dz \right) = - \left(\frac{\partial P^*}{\partial x} dx + \frac{\partial P^*}{\partial y} dy + \frac{\partial P^*}{\partial z} dz \right) \dots\dots\dots(2.34)$$

Komponen kecepatan didefinisikan $u = \frac{dx}{dt}$; $v = \frac{dy}{dt}$; dan $w = \frac{dz}{dt}$ dimana dx, dy, dan dz adalah komponen dari suatu elemen streamline dS. Kemudian turunannya dapat ditulis $dx = u dt$, $dy = v dt$, dan $dz = w dt$. Sehingga ruas kiri persamaan (2.24) dapat ditulis ke dalam bentuk

$$\rho (u du + v dv + w dw) = \rho d \left(\frac{u^2 + v^2 + w^2}{2} \right) = d \left(\frac{\rho V^2}{2} \right) \dots\dots\dots(2.35)$$

Ruas kanan dari persamaan (2.24) adalah total dari turunan P*. Jadi persamaan momentum menjadi:

$$d \left(\rho \frac{V^2}{2} \right) = -dp^* \dots\dots\dots(2.36)$$

Persamaan-persamaan di atas hanya digunakan pada suatu streamline dan diketahui bahwa $P^* = P + \rho gz$, maka hasil akhir adalah

$$\frac{d}{dS} \left(\rho \frac{V^2}{2} + P + \rho gz \right) = 0 \quad \dots\dots\dots(2.37)$$

Kemudian diintegrasikan menjadi

$$\rho \frac{V^2}{2} + P + \rho gz = \text{const} \quad \dots\dots\dots(2.38)$$



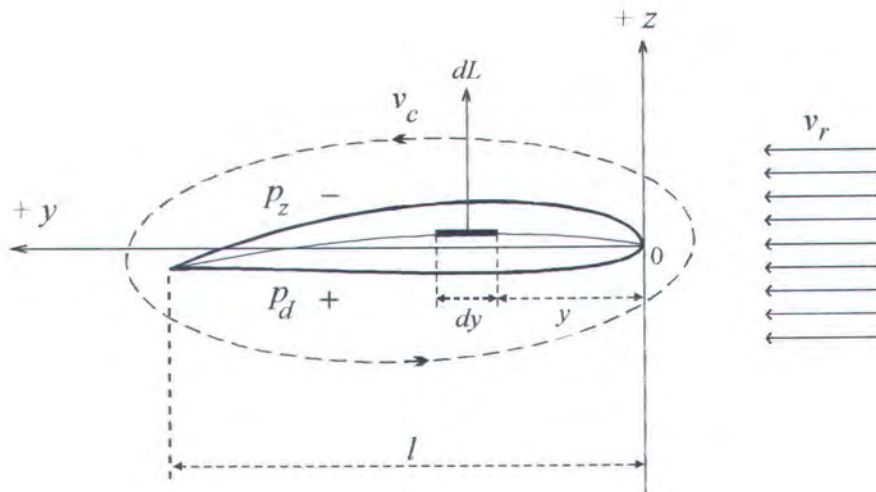
2.4. Teori Kutta-Joukowski

Teori Kutta-Joukowski dapat menjelaskan terjadinya gaya angkat pada penampang aerofoil yang berada di aliran fluida ideal. Teori tersebut menjelaskan bahwa, jika suatu benda bergerak dengan kecepatan teratur (*uniform velocity*) di dalam suatu fluida ideal maka disekitar benda tersebut akan timbul aliran sirkulasi yang menyebabkan terjadinya perbedaan tekanan. Adanya perbedaan tekanan disekitar benda, menyebabkan timbulnya gaya angkat (*lift*) pada benda tersebut. Besarnya gaya angkat dapat dinyatakan dalam persamaan:

$$dL = \rho \cdot v_r \cdot \Gamma \cdot dr \quad (\text{persamaan Kutta-Joukowski}) \quad \dots\dots\dots(2.39)$$

di mana :

- | | | | |
|--------|-------------------------------|-------|---------------------------|
| dL | : gaya angkat (<i>lift</i>) | v_r | : kecepatan aliran fluida |
| ρ | : density dari fluida | dr | : lebar elemen |



Gb. 2.11 Sirkulasi pada penampang sayap

Persamaan Kutta-Joukowski tersebut dapat dibuktikan dengan mengadakan peninjauan pada elemen daun berbentuk aerofoil (perhatikan gb. 2.11). Aerofoil mempunyai bentuk yang sedemikian hingga dapat menimbulkan aliran sirkulasi disekelilingnya. Pada bagian bawah permukaan aerofoil akan bertekanan tinggi, sehingga menurut dalil Bernoulli kecepatannya berkurang. Sedangkan pada bagian atas permukaan aerofoil akan bertekanan rendah dan kecepatannya bertambah besar. Aliran relatif sepanjang aerofoil terdiri dari:

1. Aliran *streamline* (*streamline flow*) dengan kecepatan v_r .
2. Aliran *vortex* (*vortex flow*) yang menyelubungi aerofoil dengan kecepatan v_c .

Menurut Bernoulli:

$$\begin{aligned}
 p_d + \frac{1}{2} \rho (v_r - v_c)^2 &= p_z + \frac{1}{2} \rho (v_r + v_c)^2 \\
 \Delta p &= p_d - p_z \\
 \Delta p &= 2 \cdot \rho \cdot v_r \cdot v_c \quad \rightarrow \quad \Delta v = v_c - (-v_c) = 2v_c \\
 \Delta p &= \rho \cdot v_r \cdot \Delta v
 \end{aligned}$$

di mana:

Δp : perbedaan tekanan di atas dan dibawah aerofoil

Δv : perbedaan kecepatan di atas dan dibawah aerofoil

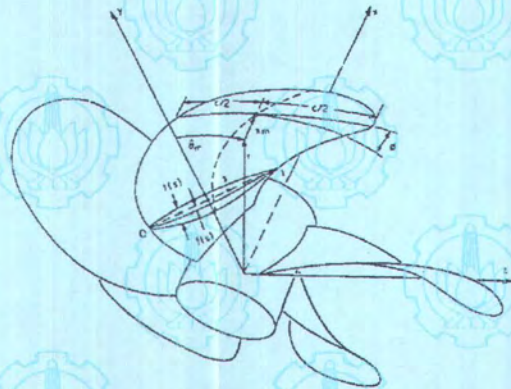
Gaya angkat dL dari aerofoil dengan panjang l dan lebar dr adalah:

$$dL = dr \cdot \int_0^l \Delta p \cdot dy = \rho \cdot v_r \cdot dr \cdot \int_0^l \Delta v \cdot dy$$

$$\Rightarrow \Gamma = \int_0^l v_c \cdot dy - \int_0^l -v_c \cdot dy = \int_0^l 2v_c \cdot dy = \int_0^l \Delta v \cdot dy$$

jadi: $dL = \rho \cdot \Gamma \cdot v_r \cdot dr \rightarrow$ persamaan Kutta-Joukowski

Dengan memakai persamaan Kutta-Joukowski tersebut dapatlah dicari besarnya gaya angkat sebuah aerofoil yang diketahui ukuran dan kecepatannya di fluida yang ditentukan, serta besarnya kekuatan sirkulasi dari aliran *vortex*-nya. Besarnya kekuatan sirkulasi (Γ) tergantung dari bentuk/penampang dari suatu aerofoil. Karena sulitnya menentukan besarnya kekuatan sirkulasi dari suatu penampang aerofoil, dalam Tugas Akhir ini tidak dibahas perhitungan mengenai harga dari kekuatan sirkulasi dari penampang aerofoil. Disini hanya disajikan beberapa hal yang perlu diketahui untuk memprediksi besarnya kekuatan sirkulasi dari suatu penampang aerofoil.



BAB III

DISTRIBUSI TEKANAN

BAB III

DISTRIBUSI TEKANAN

3.1. Metode Panel Dua Dimensi

3.1.1. Aliran Potensial sekitar Hydrofoil

Hydrofoil dilihat sebagai benda diam dan aliran uniform \bar{U}_{in} datang seperti yang ditunjukkan dalam gb. 3.1. Medan kecepatan total \bar{V} dapat ditunjukkan dengan fungsi skalar dari velocity potencial ϕ , yaitu

$$\bar{V} = \nabla \phi \quad \dots\dots\dots(3.1)$$

Persamaan continuity untuk ϕ adalah

$$\nabla^2 \phi = 0 \quad \dots\dots\dots(3.2)$$

Kondisi batas kinematik pada permukaan foil adalah

$$\frac{\partial \phi}{\partial n} = \bar{n} \cdot \nabla \phi = 0 \quad \dots\dots\dots(3.3)$$

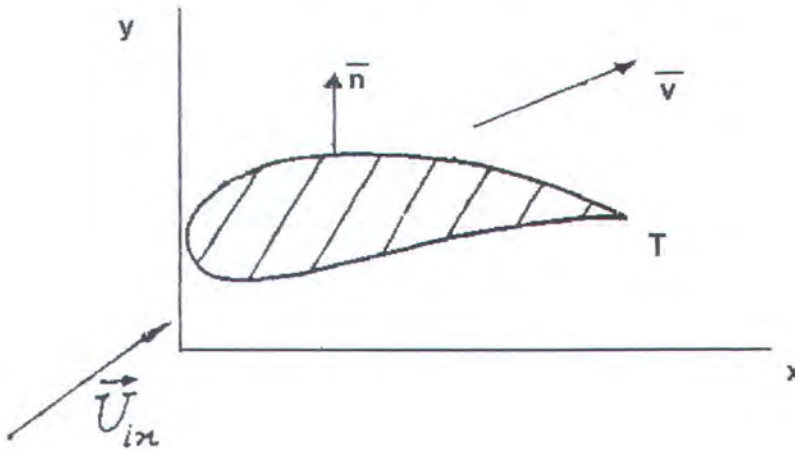
dimana \bar{n} vektor normal di permukaan foil.

Kondisi aliran bebas

$$\nabla \phi \approx \bar{U}_{in} \quad \dots\dots\dots(3.4)$$

Kondisi batas Kutta

$$\nabla \phi = \text{tertentu} ; \text{ di trailing edge T} \dots\dots\dots(3.5)$$



Gb. 3.1 Hydrofoil dalam aliran uniform

3.1.2. Perhitungan Distribusi Tekanan Secara Numerik

Metoda berikut ini adalah metoda vortex panel dua dimensi [Stevens et al., 1971]. Dalam analisa suatu benda yang dapat terangkat atau *lifting bodies* perlu diperhitungkan adanya sirkulasi maka digunakan metoda vortex panel.

Dengan metoda ini suatu bentuk hydrofoil digambarkan dengan pendekatan poligon tertutup dari vortex panel-vortex panel. Jadi ketepatan perhitungan ditentukan oleh jumlah dan panjang panel sedemikian rupa mirip bentuk hydrofoil yang akan dianalisa.

Metoda vortex panel yang dibahas di sini mempunyai asumsi bahwa kerapatan sirkulasi setiap panel berubah linear dari ujung satu ke ujung yang lain dan menerus tidak terputus di tiap ujung-ujungnya, seperti digambarkan dalam gb.3.2. Perhitungan-perhitungan dalam metoda ini juga memasukkan kondisi

batas Kutta dan perhitungan numerik akan menerus kecuali sejumlah panel dipilih untuk hydrofoil yang mempunyai sudut runcing pada trailing edge-nya. Panel-panel yang berjumlah m diasumsikan planar dan penamaan diurutkan searah jarum jam, mulai dari trailing edge. Titik batas-titik batas yang merupakan perpotongan dari vortex panel-vortex panel yang bersebelahan akan ditentukan sepanjang permukaan hydrofoil. Kondisi bahwa permukaan hydrofoil akan selalu streamline didekati dengan menerapkan kondisi komponen kecepatan normal adalah nol di setiap titik kontrol-titik kontrol, yaitu titik tengah vortex panel.

Suatu uniform flow V_∞ yang membentuk angle of attack α dan vortex panel berjumlah, velocity potential di titik kontrol ke- i (x_i, y_i) adalah :

$$\phi(x_i, y_i) = V_\infty (x_i \cos \alpha + y_i \sin \alpha) - \sum_{j=1}^m \int_j \frac{\gamma(s_j)}{2\pi} \tan^{-1} \left(\frac{y_i - y_j}{x_i - x_j} ds_j \right) \dots \dots (3.1)$$

dimana

$$\gamma(s_j) = \gamma_j + (\gamma_{j+1} - \gamma_j) \frac{s_j}{S_j} \dots \dots \dots (3.2)$$

Dari gb. 3.2, (x_j, y_j) merupakan koordinat dari titik-titik sepanjang panel ke- j , yang berjarak s_j diukur dari ujung belakang panel ke- j tersebut, dimana panjang panel ke- j itu adalah S_j . Perhitungan integral dilakukan sepanjang panel yang ditinjau yaitu dari (X_j, Y_j) ke (X_{j+1}, Y_{j+1}). Di sini huruf besar digunakan untuk memberikan notasi koordinat dari titik batas-titik batas. Harga-harga ($m + 1$) dari γ_j di titik batas adalah konstanta-konstanta yang tidak diketahui, yang harus ditentukan secara numerik.

Kondisi batas menentukan bahwa vektor kecepatan n_i yang arahnya tegak lurus (normal) keluar sama dengan nol di titik kontrol ke- i , sehingga

$$\frac{\partial}{\partial n_i} \phi(x_i - y_i) = 0 \quad i = 1, 2, 3, \dots, m$$

Penyelesaian turunan dan mengintegrasikan persamaan di atas akan didapat

$$\sum_{j=1}^m (Cn1_{ij} \gamma'_j + Cn2_{ij} \gamma'_{j+1}) = \sin(\theta_i - \alpha) \quad i = 1, 2, 3, \dots, m \quad \dots \dots (3.3)$$

yang mana $\gamma' = \frac{\gamma}{2\pi V_\infty}$ adalah kerapatan sirkulasi tak berdimensi sedang θ_i adalah

orientation angle dari panel ke- i yang diukur dari sumbu x ke permukaan panel.

Koefisien $Cn1_{ij}$ dan $Cn2_{ij}$ dirumuskan sebagai berikut

$$Cn1_{ij} = 0,5DF + CG - Cn2_{ij}$$

$$Cn2_{ij} = D + 0,5 \frac{QF}{S_j} - (AC + DE) \frac{G}{S_j}$$

Konstanta-konstanta didefinisikan sebagai berikut

$$A = -(x_i - X_j) \cos \theta_j - (y_i - Y_j) \sin \theta_j$$

$$B = (x_i - X_j)^2 + (y_i - Y_j)^2$$

$$C = \sin(\theta_i - \theta_j)$$

$$D = \cos(\theta_i - \theta_j)$$

$$E = (x_i - X_j) \sin \theta_j - (y_i - Y_j) \cos \theta_j$$

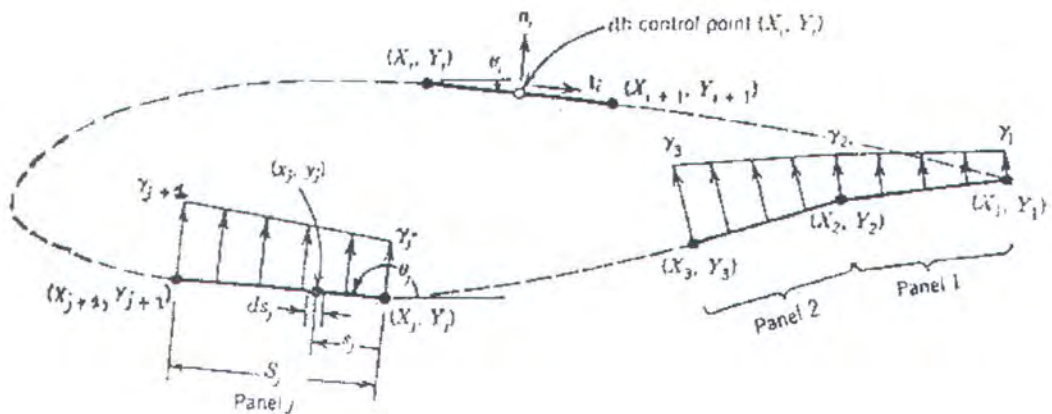
$$F = \ln \left(1 + \frac{S_j^2 + 2AS_j}{B} \right)$$

$$G = \tan^{-1} \left(\frac{ES_j}{B + AS_j} \right)$$

$$P = (x_i - X_j) \sin(\theta_i - 2\theta_j) + (y_i - Y_j) \cos(\theta_i - 2\theta_j)$$

$$Q = (x_i - X_j) \cos(\theta_i - 2\theta_j) - (y_i - Y_j) \sin(\theta_i - 2\theta_j)$$

Dari persamaan di atas dapat dilihat bahwa konstanta-konstanta ini adalah fungsi dari koordinat titik kontrol ke-i, koordinat titik batas dari vortex panel ke-j, dan orientation angle dari panel ke-j dan ke-i. Konstanta-konstanta ini dapat dihitung untuk semua harga i dan j setelah geometri panel ditentukan.



Gb. 3.2 Penempatan vortex panels dengan kekuatan vortex linier

Ekspresi dalam tanda kurung pada ruas kiri dari persamaan (3.3) menentukan kecepatan normal di titik kontrol ke-i yang disebabkan oleh distribusi vortices yang linier pada panel ke-j. Untuk $i = j$, koefisien-koefisien menjadi

$$Cn1_{ij} = -1 \quad \text{dan} \quad Cn2_{ij} = 1$$

yang menyatakan kecepatan normal yang dihasilkan sendiri di titik kontrol ke-i.

Agar separation point tepat di trailing edge, kondisi batas Kutta, $\gamma'(TE) = 0$, yaitu kekuatan vorticity di trailing edge selalu nol harus diterapkan.

$$\gamma'_1 + \gamma'_{m+1} = 0 \dots\dots\dots (3.4)$$

Ada ($m + 1$) persamaan setelah menggabungkan persamaan (3.3) dan (3.4) dan persamaan-persamaan itu cukup untuk menyelesaikan ($m + 1$) harga-harga γ'_j yang tidak diketahui. Persamaan-persamaan itu dapat dituliskan ke dalam bentuk yang lebih ringkas

$$\sum_{j=1}^{m+1} An_{ij} \cdot \gamma'_j = RHS_i \quad i = 1, 2, 3, \dots, m+1 \quad \dots\dots\dots (3.5)$$

untuk $i < m+1$:

$$An_{i1} = Cn1_{i1}$$

$$An_{ij} = Cn1_{ij} + Cn2_{ij-1} \quad j = 2, 3, 4, \dots, m$$

$$An_{im+1} = Cn2_{im}$$

$$RHS_i = \sin(\theta_i - \alpha)$$

untuk $i = m + 1$:

$$An_{i1} = An_{im+1} = 1$$

$$An_{ij} = 0 \quad j = 2, 3, 4, \dots, m$$

$$RHS_i = 0$$

Kecuali untuk $i = m + 1$, An_{ij} dapat disebut koefisien pengaruh kecepatan normal yang menentukan pengaruh dari γ'_j terhadap kecepatan normal di titik kontrol ke-i.

Setelah penentuan kerapatan sirkulasi yang tak diketahui dilakukan, proses dilanjutkan dengan menghitung kecepatan dan tekanan di titik kontrol. Di titik kontrol tersebut hanya mempunyai sebuah komponen tangensial di atas permukaan panel karena ada pernyataan bahwa komponen normal sama dengan nol. Vektor tangensial di atas panel ke- i dinotasikan dengan t_i seperti dalam gb.3.2. Jadi kecepatan lokal tak berdimensi yang didefinisikan sebagai

$$\frac{\partial \phi / \partial t_i}{V_\infty}$$



dapat dihitung, yang mempunyai persamaan

$$V_i = \cos(\theta_i - \alpha) + \sum_{j=1}^m (Ct1_{ij} \cdot \gamma'_j + Ct2_{ij} \cdot \gamma'_{j+1}) \quad i = 1, 2, 3, \dots, m \quad \dots\dots\dots (3.6)$$

dimana

$$Ct_{ij} = 0,5CF - DG - Ct2_{ij}$$

$$Ct_{ij} = C + 0,5 \frac{PF}{S_j} + (AD - CE) \frac{G}{S}$$

$$Ct1_{ii} = Ct2_{ii} = \frac{\pi}{2}$$

Ekspresi di dalam tanda kurung pada persamaan (3.6) menyatakan adanya kecepatan tangensial di titik kontrol yang ditimbulkan oleh vortices yang terdistribusi sepanjang panel ke- j . Untuk memudahkan dalam perhitungan komputer, persamaan (3.6) dapat dituliskan dalam bentuk

$$V_i = \cos(\theta_i - \alpha) + \sum_{j=1}^m At_{ij} \cdot \gamma'_j \quad i = 1, 2, 3, \dots, m \quad \dots\dots\dots (3.7)$$

dimana koefisien pengaruh kecepatan tangensial didefinisikan sebagai berikut

$$At_{i1} = Ct1_{i1}$$

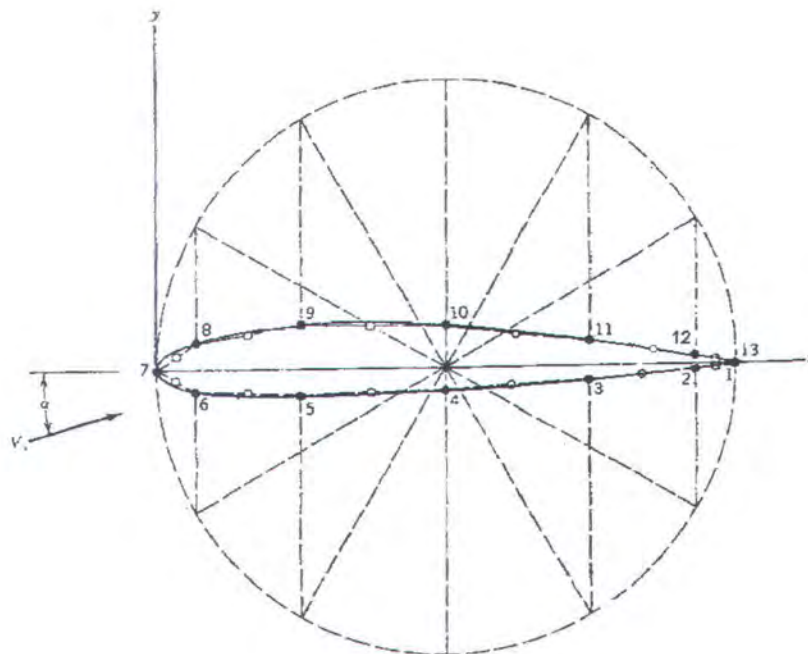
$$At_{ij} = Ct1_{ij} + Ct2_{ij-1} \quad j = 2,3,4,\dots,m$$

$$At_{im+1} = Ct2_{im}$$

Koefisien tekanan di titik kontrol ke-i adalah

$$C_p = 1 - V_i^2 \quad \dots\dots\dots(3.8)$$

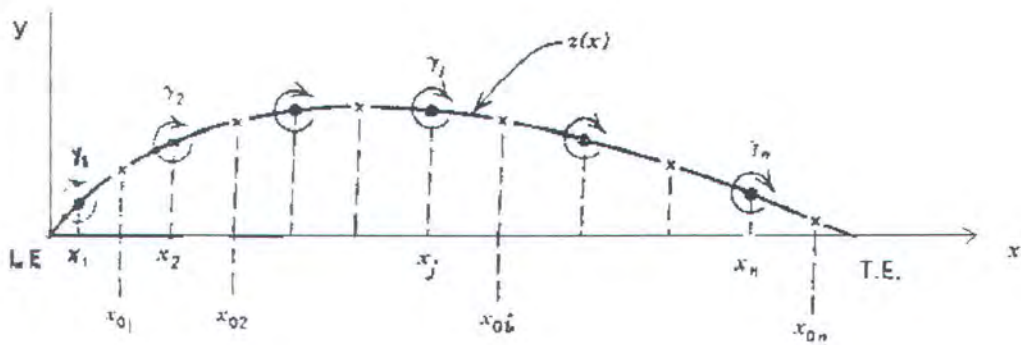
Contoh penggunaan metoda vortex panel untuk foil NACA 2412 dapat dilihat dibawah ini. Bentuk foil NACA 2412 didekati dengan menerapkan panel sebanyak 12 buah yang dipasang mengelilingi permukaan foil. Distribusi tekanan disekitar foil dapat diketahui dengan menghitung C_p tiap panel yang mempunyai 12 titik kontrol dan 13 titik batas apabila free stream datang dengan angle of attack α .



Gb. 3.3 Penentuan titik batas dan titik kontrol panel

3.2. Distribusi Tekanan dan Awal Kavitasasi

Distribusi tekanan merupakan perbandingan antara kecepatan lokal pada tiap titik kontrol dengan kecepatan pada aliran bebas. Tiap titik kontrol mempunyai nilai kecepatan yang berbeda-beda sesuai dengan posisi titik tersebut di seluruh permukaan foil. Kecepatan lokal tiap titik diketahui dari distribusi vortex sepanjang permukaan foil dan distribusi vortex dihitung berdasarkan besarnya sirkulasi pada tiap titik kontrol.



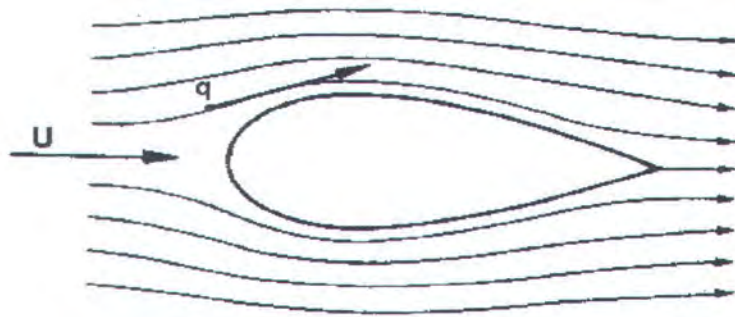
Gb. 3.4 Distribusi vortex tiap titik kontrol panel

Dari persamaan Bernoulli hubungan tekanan di tiap titik di permukaan foil dengan tekanan di titik dalam aliran bebas adalah

$$P_{\infty} + \frac{1}{2} \rho U^2 = P_1 + \frac{1}{2} \rho V_1^2 \quad \dots\dots\dots (3.9)$$

dimana V_1 adalah nilai dari kecepatan fluida total di salah satu titik di permukaan foil

$$V_1 \equiv \sqrt{(U + u_1)^2 + v_1^2}$$



Gb. 3.5 Kecepatan fluida yang terganggu oleh foil

Titik A pada aliran streamline mempunyai hubungan sebagai berikut

$$P_{\infty} + \frac{1}{2} \rho U^2 = P_1 + \frac{1}{2} \rho V_1^2 \quad \dots\dots\dots(3.10)$$

dan perubahan tekanan akan

$$\begin{aligned} \delta P &= P_1 - P_{\infty} \\ \delta P &= \frac{1}{2} \rho [U^2 - V_1^2] \quad \dots\dots\dots(3.11) \end{aligned}$$

dimana : P_{∞} = tekanan aliran bebas P_1 = tekanan di titik A
 U = kecepatan aliran bebas V_1 = kecepatan di titik A

Jika V_1 lebih besar dari U yaitu aliran mengalami percepatan maka P_1 akan lebih kecil dari P_{∞} atau tekanan mengalami penurunan dan perubahan tekanan δP adalah negative, begitu sebaliknya.

Pada titik tertentu S dekat dengan nose foil aliran akan terbagi dan fluida yang merupakan streamline pemisah menabrak permukaan foil secara tegak lurus. Fluida ini akan kehilangan seluruh kecepatan dan momentumnya sehingga di titik S kecepatan V adalah nol dan

$$\delta P = P_1 - P_{\infty} = \frac{1}{2} \rho U^2 \quad \dots\dots\dots(3.12)$$

Tekanan yang meningkat tinggi di titik S akan menyamai tekanan P_∞ . Titik S disebut stagnation point, diberi simbol q. Jadi

$$q = \frac{1}{2} \rho U^2 \quad \dots\dots\dots(3.13)$$

Fluida yang berada di atas streamline pemisah akan mengalir melampaui permukaan atas foil atau back dengan penambahan kecepatan dan menghasilkan penurunan tekanan. Sementara fluida yang mengalir melampaui permukaan bawah foil atau face akan mengalami penurunan kecepatan sehingga tekanan akan meningkat.

Di suatu titik pada sisi back dimana kecepatan dan tekanan adalah V_1 dan P_1 maka

$$\begin{aligned} P_1 &= P_\infty + \frac{1}{2} \rho [U^2 - V_1^2] \\ P_1 &= P_\infty + \delta P \end{aligned} \quad \dots\dots\dots(3.14)$$

Tekanan P_1 akan menjadi nol jika

$$\delta P = -P_\infty \quad \dots\dots\dots(3.15)$$

Ini berarti bahwa aliran akan patah, mengingat bahwa air tidak dapat lagi mempertahankan tegangannya sehingga gelembung dan rongga udara akan mulai terjadi. Fenomena alam ini disebut kavitasi.

$$P_v = \Delta P + P_\infty \quad \dots\dots\dots(3.16)$$

dimana P_v adalah tekanan uap air pada saat air mulai mendidih. Kavitasi akan terjadi bila

$$-\Delta P > P_\infty - P_v \quad \dots\dots\dots(3.17)$$

Untuk mempermudah dalam menganalisa maka tekanan di sekitar foil dinyatakan dalam bentuk tak berdimensi yaitu koefisien tekanan C_p . C_p adalah

perbandingan antara perbedaan tekanan di permukaan foil dan aliran bebas dengan tekanan dinamik aliran bebas, yaitu

$$C_p = \frac{P_1 - P_\infty}{\frac{1}{2} \rho U^2} = 1 - \left(\frac{V_1}{U} \right)^2 \dots\dots\dots(3.18)$$

Awal kavitasi dapat dicari dengan membandingkan harga minimum dari koefisien tekanan di permukaan foil dengan harga angka kavitasi, tak berdimensi yang dirumuskan sebagai berikut

$$\sigma = \frac{P_\infty - P_v}{\frac{1}{2} \rho U^2} \dots\dots\dots(3.19)$$

dimana P_v adalah tekanan uap fluida saat temperatur tertentu. Dengan membandingkan definisi dari C_p dan σ , jelas bahwa jika $-C_p > \sigma$ maka $P_1 < P_v$.

Awal kavitasi adalah $-(C_p)_{\min} = \sigma$.

3.3. Gaya Angkat

Apabila fluida sejati mengalir sekitar benda tercelup atau apabila sebuah benda bergerak dalam fluida diam, ada dua gaya yang bekerja. Gaya-gaya itu adalah gaya normal yang menyebabkan tekanan dan gaya geser yang menyebabkan geseran viskous. Dalam tugas akhir ini dibatasi benda dalam fluida ideal sehingga hanya tekanan saja yang terjadi akibat adanya gaya normal.

Gejala tentang gaya angkat yang terjadi dalam sebuah fluida ideal akibat penambahan sebuah free vortex di seputar sebuah silinder dalam suatu aliran rektilinear telah diuraikan pada bab II yang dikenal efek Magnus. Gaya

angkat untuk mengangkat bidang angkat umumnya didefinisikan dengan persamaan

$$L = C_L \frac{1}{2} \rho U^2 A \quad \dots\dots\dots(3.20)$$

dengan C_L koefisien gaya angkat, $\frac{1}{2} \rho U$ tekanan dinamik arus bebas, dan A luas chord bidang angkat, sedangkan untuk kasus dua dimensi adalah

$$L = C_L \frac{1}{2} \rho U^2 c \quad \dots\dots\dots(3.21)$$

dimana c adalah panjang chord.

Menurut persamaan Bernoulli yang sudah dijelaskan dalam subbab 2.2.3 tentang silinder dengan sirkulasi

$$L = -\int_0^{2\pi} (P - P_\infty) \sin \theta (R d\theta)$$

Tekanan relatif juga dapat dihitung dari C_p seperti dalam subbab 3.2. tentang konsep distribusi tekanan

$$P - P_\infty = C_p \frac{1}{2} \rho U^2$$

dimana C_p adalah koefisien tekanan yang dapat dicari dengan perhitungan numerik dalam subbab 3.1.2. tentang perhitungan dengan metoda vortex panel dua dimensi.

Kemudian integral $P - P_\infty$ adalah gaya angkat sehingga gaya angkat total adalah

$$L_{tot} = \int P - P_\infty = \int C_p \frac{1}{2} \rho U^2 \sin \theta \quad \dots\dots\dots(3.22)$$

sedangkan apabila gaya angkat dihitung per panel adalah

$$L = C_p \frac{1}{2} \rho U^2 \sin \theta \quad \dots\dots\dots(3.23)$$

di sini $\sin \theta$ menunjukkan proyeksi gaya normal yang merupakan penyebab tekanan itu ada ke sumbu tegak lurus aliran bebas.

Kemudian koefisien gaya angkat per panel adalah

$$C_L = \frac{C_p \frac{1}{2} \rho U^2 \sin \theta}{\frac{1}{2} \rho U^2 x} \dots\dots\dots(3.24)$$

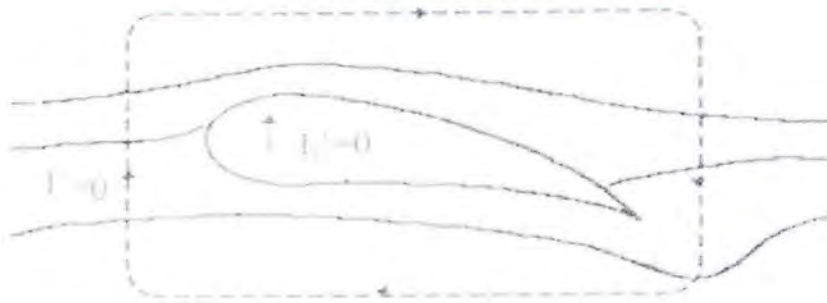
$$C_L = \frac{C_p \sin \theta}{x}$$

3.4. Kondisi Kutta

Teorema Kutta-Joukowski menyatakan bahwa gaya yang dialami oleh bodi dalam aliran uniform adalah sama dengan hasil kali dari kerapatan fluida, kecepatan aliran, dan sirkulasi dan mempunyai arah tegak lurus terhadap kecepatan aliran. Ada satu aliran tanpa rotasi dapat ditemukan yang memenuhi kondisi batas pada aliran tak hingga dan pada bodi, asalkan sirkulasi ditentukan lebih dulu. Jika sirkulasi tidak ditentukan, kondisi pada aliran tak hingga dan geometri dari bodi tidak dapat menentukan garis aliran.

Kondisi Kutta adalah kondisi sebuah penampang dengan trailing edge yang tajam dalam sebuah gerakan fluida yang berupa sirkulasi yang cukup kuat untuk menempatkan titik stagnasi pada trailing edge.

Aliran di sekitar penampang dengan suatu angle of attack dalam aliran inviscid tidak menimbulkan sirkulasi dan titik stagnasi di permukaan atas penampang.



Gb. 3.6 Aliran sekitar penampang dengan sirkulasi nol

3.5. Cavitation Buckets Diagram

Pada pembahasan lalu telah dilakukan perhitungan distribusi tekanan di sekitar foil dengan metode vortex panel dua dimensi [Stevens et al, 1971]. Metode ini dapat digunakan untuk mengetahui awal kavitasi dengan mencari tekanan minimum di permukaan foil dan membandingkannya dengan tekanan uap. Dalam hal tak dimensi, awal kavitasi dapat dilihat dari nilai $-C_p$ min dan σ .

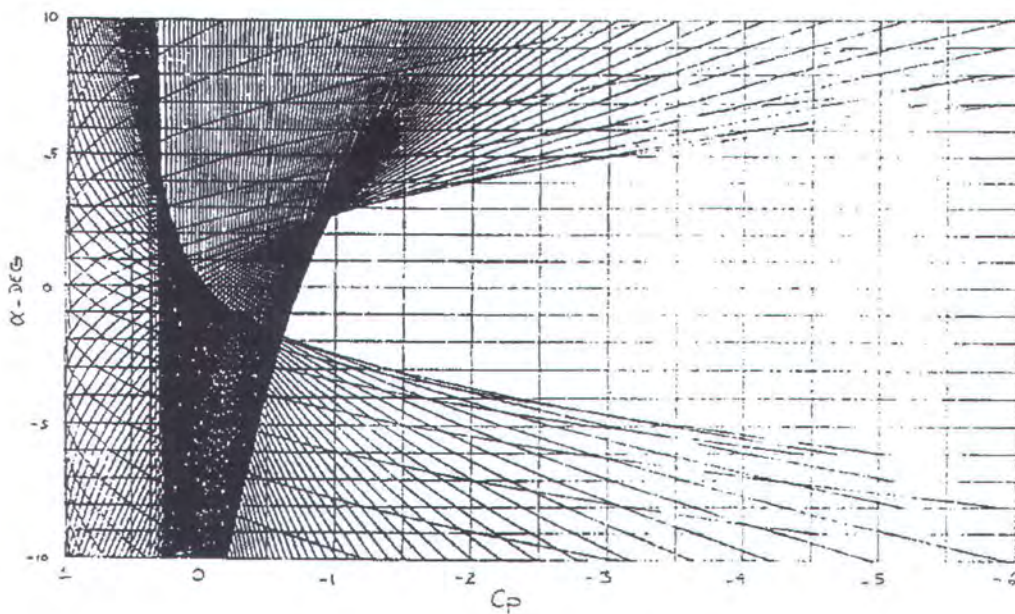
Awal kavitasi dapat dianalisa dalam suatu *cavitation buckets diagram*. Ada dua bentuk untuk diagram ini, pertama adalah variasi angle of attack lawan $-C_p$ min dan kedua adalah variasi koefisien lift C_L lawan $-C_p$ min. Agar angka kavitasi langsung diketahui maka $-C_p$ min dapat diganti σ . Satu buah *bucket diagram* menunjukkan batas tekanan minimum suatu foil dengan $\frac{1}{c}$ yang diketahui untuk menghindari terjadinya kavitasi. Kemudian dari suatu foil chamber NACA yang mempunyai distribusi tebal dan chamber ratio dapat dibentuk satu set *bucket diagram* yang masing-masing *bucket diagram*

menentukan batas tekanan minimum untuk γ_c yang berbeda, perbedaan γ_c dilakukan dengan mengubah tebal foil dan chord length. Di sini variasi angle of attack adalah antara -5 derajat sampai $+6$ derajat sedang variasi $-C_p$ min atau angka kavitasi σ adalah antara 0 sampai 3.

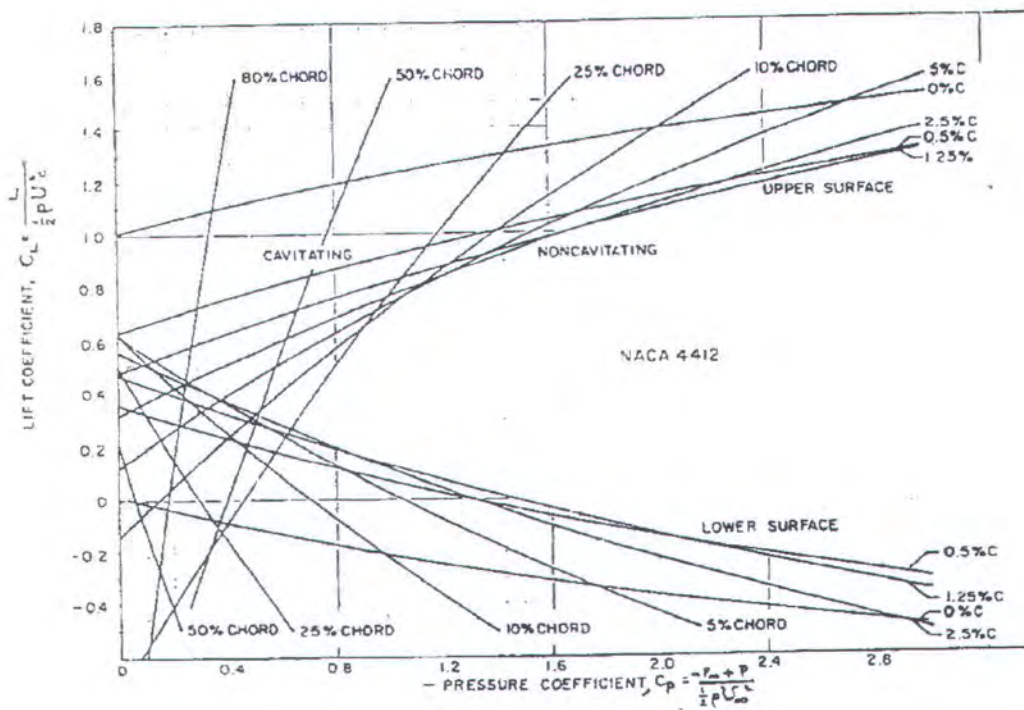
Untuk mengetahui awal kavitasi, tidak perlu untuk mengetahui seluruh tekanan di permukaan foil, cukup tekanan minimum pada angle of attack tertentu. Satu *bucket diagram* dibentuk oleh dua bagian, yaitu batas $-C_p$ min untuk leading edge sisi hisap (back) dan batas $-C_p$ min untuk leading edge sisi tekan (face). Tekanan minimum sisi tekan (face) dibentuk berdasar variasi angle of attack negatif sedang sisi hisap dibentuk berdasar variasi angle of attack positif, bahkan sampai negatif bila γ_c semakin besar.

Dengan metode panel dua dimensi, menghitung tekanan minimum ($-C_p$ min) di sisi hisap untuk suatu foil chamber dengan γ_c tertentu yaitu foil dimiringkan dengan angle of attack $6^\circ, 5^\circ, 4^\circ, 3^\circ, 2^\circ, 1^\circ, 0^\circ, -1^\circ$ dan seterusnya sampai bertemu dengan garis $-C_p$ min di sisi tekan dan tekanan minimum untuk tiap angle of attack diketahui. Nilai-nilai $-C_p$ min diplotkan dalam *cavitation buckets diagram* kemudian ditarik garis untuk menghubungkan titik-titik itu. Sedang untuk menghitung tekanan minimum di sisi tekan dapat dilakukan dengan cara yang sama yaitu foil dimiringkan dengan angle of attack $-5^\circ, -4^\circ, -3^\circ, -2^\circ, -1^\circ, 0^\circ$ atau sampai bertemu dengan batas $-C_p$ min di sisi hisap.

Satu buah *bucket diagram* [Kerwin, 1993] juga dapat dibentuk dengan menghitung seluruh tekanan di sisi hisap dan di sisi tekan suatu foil chamber. Perhitungan dipisahkan antara sisi hisap dan sisi tekan. Tekanan dihitung untuk semua angle of attack positif dan negatif untuk setiap γ_c suatu foil di sisi hisap, begitu juga untuk semua angle of attack negatif untuk setiap γ_c sehingga terbentuk garis-garis batas $-C_p$ min untuk γ_c tertentu dengan parameter γ_c . Grafik-grafik yang terdalam akan membentuk sebuah bentuk ember kavitasi atau cavitation bucket seperti dalam gb. 3.2. Perhitungan yang sama juga dapat dilakukan untuk menganalisa koefisien gaya angkat C_L seperti dalam gb. 3.3.



Gb. 3.7 Koefisien tekanan lawan angle of attack



Gb. 3.8 Koefisien tekanan lawan koefisien gaya angkat

Bucket diagram merupakan awal kavitasi (inception of cavitation) dari sisi hisap dan sisi tekan suatu foil. Jadi foil yang didisain tidak terjadi kavitasi bila $-C_{pmin}$ -nya di dalam *bucket diagram*-nya sesuai dengan $\frac{1}{c}$ -nya. Bentuk dari setiap bucket diagram menentukan jangkauan koefisien gaya angkat atau angle of attack. Jangkauan ini adalah batasan kavitasi dimana akan dipilih suatu nilai dengan tebal tertentu dalam spiral design.

Dalam gb. 3.9 dari angka kavitasi, adalah angka yang tergantung dari kondisi lingkungan, yang diketahui ditarik garis tegak memotong bucket diagram di sisi tekan dan sisi hisap suatu $\frac{1}{c}$. Selisih ordinat atas, koefisien gaya angkat di sisi hisap, dan ordinat bawah, koefisien di sisi tekan adalah variasi maksimum dari koefisien gaya angkat atau jangkauan ΔC_L tanpa adanya kavitasi. Bila ordinat itu

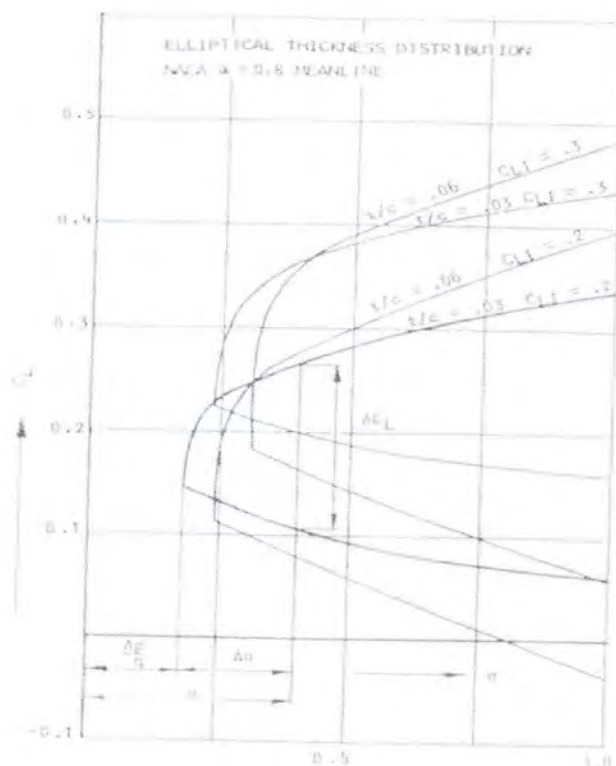
adalah α maka selisih itu menghasilkan variasi dari angle of attack tanpa adanya kavitasi.

Nilai batas $-C_{pmin}$ agar terhindar dari *mid-chord cavitation* adalah selisih $-C_{pmin}$ antara angka kavitasi dan *bucket diagram* [Gunsteren dan Pronk, 1977]

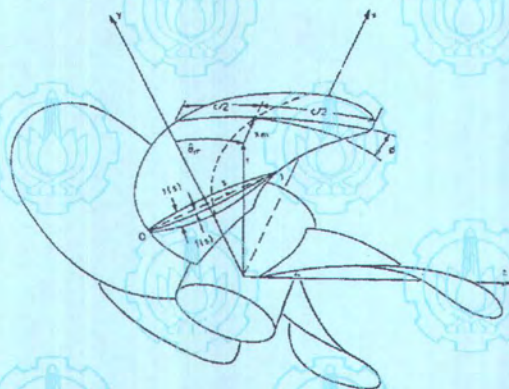
$$\Delta\sigma = \sigma - \frac{\Delta p}{q}$$

dimana $\frac{\Delta p}{q}$ adalah shock-free entry atau $-C_{pmin}$ dalam gb. 3.9.

Mid-chord cavitation terjadi bila $\Delta\sigma$ berharga negatif dan hal ini sangat berbahaya terhadap hydrofoil. Jadi nilai positif dari $\Delta\sigma$ adalah merupakan keharusan dalam setiap disain.



Gb. 3.9 Geometri dan koefisien gaya angkat pada angka kavitasi



BAB IV

ANALISA HASIL RUNNING

PROGRAM

BAB IV

ANALISA HASIL RUNNING PROGRAM

4.1. Algoritma Program

4.1.1. Algoritme Pembentukan Foil NACA Seri 4 Digit

Dalam pembuatan Sketsa Foil NACA 4 Digit ditempuh dua tahap pembentukan koordinat, yaitu: membentuk koordinat mean line dan koordinat ketebalan foil (dengan menggunakan 2 buah parameter f/c dan t/c). Algoritma programnya adalah sebagai berikut:

```

procedure TFoil.SetFoil;
var
  i:integer;
begin
  for i:=0 to High(XT) do
  begin
    XT[i]:=i/High(XT);
    YT[i]:=(TC*(1.4845*sqrt(XT[i])-0.63*XT[i]-
    1.758*sqrt(XT[i])+1.4215*Power(XT[i],3)-0.5075*Power(XT[i],4)));
  end;
  for i:=0 to High(XT) do
  if(XT[i]<=XFC)Then
  begin
    YC[i]:=(FC/SQR(XFC))*(2*XFC*XT[i]-SQR(XT[i]));
    THETAC[i]:=ArcTan((FC/SQR(XFC))*2*(XFC-XT[i]));
  end

```

```

else
begin
YC[i]:=(FC/SQR(1-XFC))*(1-2*XFC+2*XFC*XT[i]-SQR(XT[i]));
THETAC[i]:=ArcTan((FC/SQR(1-XFC))*2*(XFC-XT[i]));
end;
for i:=0 to High(XT) do
begin
XS[i]:=XT[High(XT)-i]+YT[High(XT)-i]*Sin(THETAC[High(XT)-i]);
YS[i]:=YC[High(XT)-i]-YT[High(XT)-i]*Cos(THETAC[High(XT)-i]);
end;
for i:=1 to High(XT) do
begin
XS[High(XT)+i]:=XT[i]-YT[i]*Sin(THETAC[i]);
YS[High(XT)+i]:=YC[i]+YT[i]*Cos(THETAC[i]);
end;
XS[0]:=1;
XS[High(XS)]:=1;
YS[0]:=0;
YS[High(XS)]:=0;
end;

```

4.1.2. Algoritma Pengubahan ketebalan Foil

Foil yang telah terbentuk ditebal-tipiskan tanpa secara proporsional dengan memanipulasi besarnya f/c dan atau t/c. Algoritma programnya adalah sebagai berikut:

- **Pengubahan f/c**

```

Constructor TNACA.NACA(SERIAL:String;m:integer);
var
  i:integer;
  YM,XFC,FC:Double;
begin
SERIES:=Copy(SERIAL,1,LENGTH(SERIAL));
PANELCOUNT:=m;
Val(Copy(SERIES,2,1),XFC,i);
XFC:=XFC/10;
Val(Copy(SERIES,3,2),YM,i);
YM:=YM/100;
Val(Copy(SERIES,1,1),FC,i);
FC:=FC/100+0.008;
for i:=0 to 2 do
begin

```

```

FC:=FC-0.004;
MLine[i]:=TMeanLines.MeanLines(PANELCOUNT,FC,XFC,YM);
end;
end;

```

- **Pengubahan t/c**

```

Constructor TMeanlines.Meanlines(m:integer;TFC,TXFC,TTC:Double);
var
  i:integer;
  TC:Double;
begin
  PANELCOUNT:=m;
  FC:=TFC;
  TC:=TTC+0.025;
  for i:=0 to 9 do
  begin
    TC:=TC-0.005;
    PFoil[i]:=TFoil.Foil(PANELCOUNT,FC,TXFC,TC);
  end;
end;

```

4.1.3. Algoritme Perhitungan Distribusi Tekanan Foil.

Penentuan koordinat Batas Panel (XB,YB)

```

procedure TFoil.SetPanelBoundary;

var

  i,j:integer;

  ANGLE:Double;

begin

  XB:=VarArrayCreate([1,PANELCOUNT+1],varDouble);

  YB:=VarArrayCreate([1,PANELCOUNT+1],varDouble);
  for i:=0 to PANELCOUNT do
  begin
    ANGLE:=(i/PANELCOUNT)*360;
    j:=FindPoint(ANGLE);
    XB[i+1]:=XS[j];
    YB[i+1]:=YS[j];
  end;
end;

```


Penentuan Parameter Panel (Titik Kontrol Panel, Panjang busur Panel, dan Sudut panel)

```

procedure TFoil.SetPanelParameter;
var
  i,PL1:integer;
begin
  XY:=VarArrayCreate([1,SumPanel,1,2],varDouble);
  THETA:=VarArrayCreate([1,SumPanel],varDouble);
  S:=VarArrayCreate([1,SumPanel],varDouble);
  for i:=1 to SumPanel do

    begin
      PL1:=i+1;
      XY[i,1]:=0.5*(XYB[i,1]+XYB[PL1,1]);
      XY[i,2]:=0.5*(XYB[i,2]+XYB[PL1,2]);
      THETA[i]:=ARCTAN2((XYB[PL1,2]-XYB[i,2]),(XYB[PL1,1]-
        XYB[i,1]));
      S[i]:=sqrt(SQR(XYB[PL1,1]-XYB[i,1])+SQR(XYB[PL1,2]-
        XYB[i,2]));
    end;
  end;
end;

```

- **Perhitungan Distribusi Tekanan pada Panel**

```

procedure TFoil.SetDistribution(r:Double);

var

  i,j:integer;

  A,B,C,D,E,F,G,P,Q:Double;

  AN,AT,CN1,CN2,CT1,CT2,SINE,COSINE,RHS,TEMPCP:Variant;

begin

  Attack:=DegToRad(r);
  SetPanelParameter;
  SINE:=VarArrayCreate([1,SumPanel],varDouble);
  COSINE:=VarArrayCreate([1,SumPanel],varDouble);
  for i:=1 to SumPanel do
    begin
      SINE[i]:=SIN(THETA[i]);

```

```

    COSINE[i]:=COS(THETA[i]);
end;
RHS:=VarArrayCreate([1,SumPanel+1],varDouble);
for i:=1 to SumPanel do
    RHS[i]:=SIN(THETA[i]-Attack);
    RHS[SumPanel+1]:=0;
    GAMMA:=VarArrayCreate([1,SumPanel+1],varDouble);
    VP:=VarArrayCreate([1,SumPanel],varDouble);
    CP:=VarArrayCreate([1,SumPanel],varDouble);
    CN1:=VarArrayCreate([1,SumPanel,1,SumPanel],varDouble);
    CN2:=VarArrayCreate([1,SumPanel,1,SumPanel],varDouble);
    CT1:=VarArrayCreate([1,SumPanel,1,SumPanel],varDouble);
    CT2:=VarArrayCreate([1,SumPanel,1,SumPanel],varDouble);
    AN:=VarArrayCreate([1,SumPanel+1,1,SumPanel+1],varDouble);
    AT:=VarArrayCreate([1,SumPanel,1,SumPanel+1],varDouble);
    for i:=1 to SumPanel do
        for j:=1 to SumPanel do
            if(i=j)then
                begin
                    CN1[i,j]:=-1;
                    CN2[i,j]:=1;
                    CT1[i,j]:=0.5*PI;
                    CT2[i,j]:=0.5*PI;
                end
            else
                begin
                    A:=(XYB[j,1]-XY[i,1])*COSINE[j]-(XY[i,2]-XYB[j,2])*SINE[j];
                    B:=SQR(XY[i,1]-XYB[j,1])+SQR(XY[i,2]-XYB[j,2]);
                    C:=Sin(THETA[i]-THETA[j]);
                    D:=Cos(THETA[i]-THETA[j]);
                    E:=((XY[i,1]-XYB[j,1])*SINE[j]-(XY[i,2]-XYB[j,2])*COSINE[j]);
                    F:=Ln(1+((SQR(S[j])+2*A*S[j])/B));
                    G:=ArcTan2(E*S[j],B+A*S[j]);
                    P:=(XY[i,1]-XYB[j,1])*SIN(THETA[i]-2*THETA[j])+(XY[i,2]-
                        YB[j,2])*COS(THETA[i]-*THETA[j]);
                    Q:=(XY[i,1]-XYB[j,1])*COS(THETA[i]-2*THETA[j])-(XY[i,2]-
                        YB[j,2])*SIN(THETA[i]-*THETA[j]);
                    CN2[i,j]:=D+0.5*Q*F/S[j]-(A*C+D*E)*G/S[j];
                    CN1[i,j]:=0.5*D*F+C*G-CN2[i,j];
                    CT2[i,j]:=C+0.5*P*F/S[j]+(A*D-C*E)*G/S[j];
                    CT1[i,j]:=0.5*C*F-D*G-CT2[i,j];
                end;
            end;
        for i:=1 to SumPanel do
            begin
                AN[i,1]:=CN1[i,1];
                AN[i,SumPanel+1]:=CN2[i,SumPanel];
            end;
        end;
    end;

```

```

    AT[i,1]:=CT1[i,1];
    AT[i,SumPanel+1]:=CT2[i,SumPanel];
    for j:=2 to SumPanel do
    begin
    AN[i,j]:=CN1[i,j]+CN2[i,j-1];
    AT[i,j]:=CT1[i,j]+CT2[i,j-1];
    end;
    end;
    AN[SumPanel+1,1]:=1;
    AN[SumPanel+1,SumPanel+1]:=1;
    for j:=2 to SumPanel do
    AN[SumPanel+1,j]:=0;
    GAMMA:=Gaussian(AN,RHS);
    for i:=1 to SumPanel do
    begin
    VP[i]:=Cos(THETA[i]-Attack);
    for j:=1 to SumPanel+1 do
    begin
    VP[i]:=VP[i]+AT[i,j]*GAMMA[j];
    CP[i]:=1-SQR(VP[i]);
    end;
    end;
    TEMPCP:=VarArrayCreate([1,SumPanel+1,1,2],varDouble);
    for i:=1 to SumPanel do
    begin
    TEMPCP[i,1]:=XYB[i,1];
    TEMPCP[i,2]:=CP[i];
    end;
    CPMIN:=CP[FindIndexMin(2,TEMPCP)];
    end;

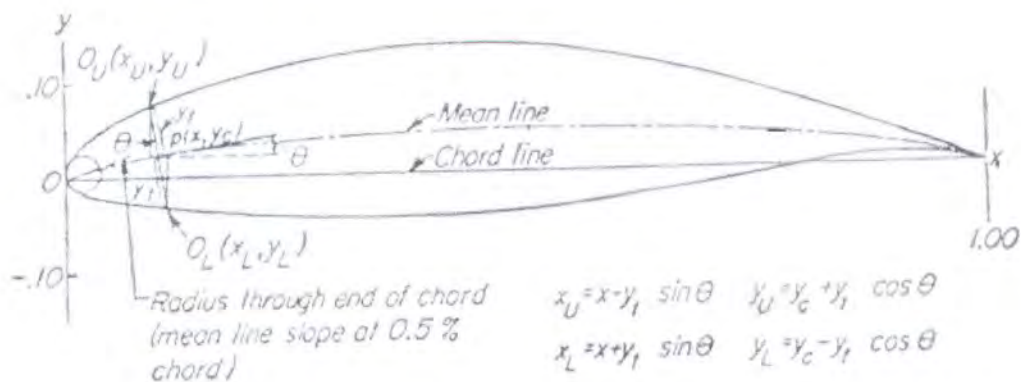
```

4.2. Perubahan Thickness Ratio dan Camber Ratio

Disain foil pada preliminary design menghasilkan geometri foil yang belum tetap, masih mengalami perubahan seiring perencanaannya dalam spiral design. Perubahan itu adalah optimisasi geometri foil hasil disain. Analisa kavitasi dalam *cavitation buckets diagram* menyediakan diagram pengaruh perubahan

thickness ratio dan camber ratio dalam mendisain suatu foil. Foil yang digunakan di sini adalah foil NACA *four-digit*.

Foil NACA *four-digit* adalah foil camber. Geometri foil camber itu merupakan gabungan geometri foil simetri dan mean line. Mean line adalah garis acuan untuk menentukan distribusi tebal sepanjang chord line. Distribusi tebal yang membentuk foil NACA camber merupakan distribusi tebal foil NACA simetri. Jadi foil camber adalah foil yang mempunyai distribusi tebal atas dan bawah sama terhadap mean line. Pembentukan foil camber dapat dilihat dalam gambar di bawah ini



Gb. 4.1 Metode kombinasi mean line dan basic thickness form

Permukaan atas dan bawah foil camber didapat dengan meletakkan tebal foil simetri di atas mean line secara tegak lurus. Kemudian koordinat-koordinat permukaan atas dan bawah terhadap chord line adalah sebagai berikut :

- Koordinat permukaan atas

$$\begin{aligned} X_U &= x - y_t \sin \theta \\ Y_U &= y_c + y_t \cos \theta \end{aligned}$$

- Koordinat permukaan bawah $X_U = x + y_t \sin \theta$
 $Y_U = y_c - y_t \cos \theta$

Dimana x , y_c , dan $\tan \theta$ adalah absis, ordinat, dan slope dari mean line sedangkan y_t adalah distribusi tebal foil sepanjang chord. Bila titik-titik koordinat foil dihubungkan maka akan mendapatkan bentuk foil yang tidak simetri.

Data yang dibutuhkan untuk membentuk suatu foil camber ditampilkan dalam Appendix I untuk distribusi tebal foil simetri (basic thickness form) dan Appendix II untuk ordinat mean line. Koordinat permukaan foil ditampilkan dalam Appendix III dari buku Theory of Wing Section.

Distribusi tebal dari penampang NACA *four-digit* mempunyai persamaan sebagai berikut :

$$\pm y_t = \frac{t}{0,2} (0,2969\sqrt{x} - 0,1260x - 0,3516x^2 + 0,2843x^3 - 0,1015x^4)$$

dimana t adalah tebal maksimum dalam per cent dari chord. Besarnya radius leading-edge adalah :

$$r_t = 1,1019.t^2$$

Ordinat untuk masing-masing thickness ratio 6, 9, 12, 15, 18, 21, dan 24 per cent diberikan oleh Appendix I.

Kelengkungan mean line mempunyai dua persamaan yang masing-masing ditentukan oleh letak ordinat maksimum di chord line :

$$y_c = \frac{m}{p^2} (2px - x^2) \dots\dots\dots \text{ordinat}$$

depan

$$y_c = \frac{m}{(1-p)^2} [(1-2p) + 2px - x^2] \quad \dots\dots\dots \text{ordinat}$$

belakang

dimana :

m = ordinat maksimum mean line dalam per cent dari chord.

p = letak ordinat maksimum di chord line dari leading-edge.

Data yang menampilkan lengkung mean line dengan ordinat maksimum 6 per cent dari chord dan letaknya dari leading-edge adalah 20, 30, 40, 50, 60, dan 70 per cent dari chord diberikan oleh Appendix II.

NACA *four-digit* mempunyai sistem nomor yang dapat mendiskripsikan geometri dari NACA itu sendiri. Digit pertama menandai harga maksimum ordinat mean line y_c dalam per cent dari chord. Digit kedua menandai letak dari ordinat maksimum mean line x_c di chord line yang diukur dari leading-edge dalam persepuluh dari chord. Dua digit terakhir menandai thickness ratio $\frac{t}{c}$ dalam per cent dari chord. Jadi NACA 4412 mempunyai 4 per cent tinggi camber di 0,4 dari leading-edge dan tebalnya adalah 15 per cent atau dengan kata lain NACA 4412 dapat dipisahkan menjadi dua yaitu NACA 44 mean line dan NACA 0012 basic thickness form.

4.3. Pengaruh Perubahan Thickness Ratio terhadap Angle of Attack dan Koefisien Gaya Angkat

Distribusi tebal dari foil NACA akan diubah dengan menebalkan dan menipiskan geometri foil secara proporsional yaitu tidak mengubah posisi tebal maksimumnya. Misalnya untuk foil NACA 4412 akan menjadi NACA 4402,

NACA 4404, NACA 4408 bila ditipiskan dan NACA 4416, NACA 4420 bila ditebalkan. Geometri foil dapat dibentuk dengan persamaan distribusi tebal yang sudah disebutkan dalam subbab 4.2.

Satu foil NACA yang disebutkan di atas, menandai thickness ratio $\%$ tertentu, membentuk satu *bucket diagram*. Kemudian bila semua foil NACA dimasukkan dalam perhitungan maka didapat satu set *cavitation buckets diagram*, mengenai pembentukan *bucket diagram* sudah disebutkan dalam subbab 3.5.

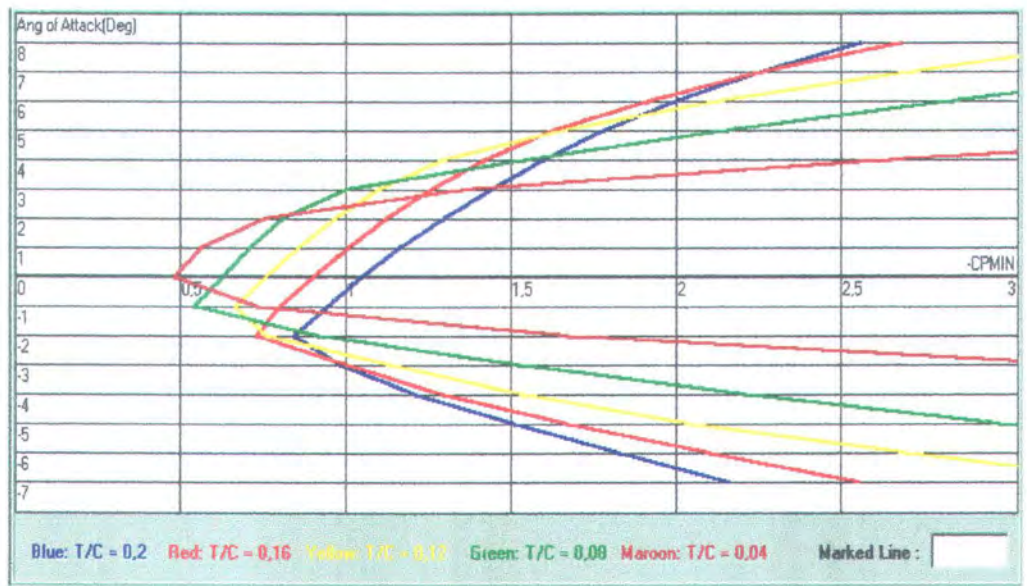
Running program menghasilkan data yang ditampilkan dalam tabel 4.1, merupakan data $-C_{p\min}$ untuk setiap perubahan angle of attack untuk membentuk satu *bucket diagram*. Kemudian tabel-tabel itu membentuk satu set *cavitation buckets diagram* yang ditampilkan dalam gb. 4.2.

Perubahan tebal menghasilkan bucket diagram-bucket diagram yang berbeda. Perubahan itu meliputi lebar dan dalam. Semakin tebal suatu foil, bucket yang terbentuk semakin lebar tetapi semakin dangkal. Nilai dari panjang-dalamnya suatu bucket tergantung dari sumbu ordinat, yaitu angle of attack α atau koefisien gaya angkat C_L .

tabel 4.1

$FC=0,04$

$T/C=0,04$		$T/C=0,08$		$T/C=0,12$		$T/C=0,16$		$T/C=0,2$	
ALPHA	-CPMIN	ALPHA	-CPMIN	ALPHA	-CPMIN	ALPHA	-CPMIN	ALPHA	-CPMIN
-7	2,163	-7	2,557	-7	3,404	-7	5,599	-7	12,86
-6	1,829	-6	2,102	-6	2,694	-6	4,21	-6	9,937
-5	1,512	-5	1,676	-5	2,044	-5	2,981	-5	7,353
-4	1,215	-4	1,296	-4	1,54	-4	2,211	-4	5,113
-3	0,9835	-3	1,005	-3	1,134	-3	1,523	-3	3,219
-2	0,8455	-2	0,7321	-2	0,7618	-2	0,9165	-2	1,673
-1	0,9442	-1	0,8025	-1	0,6573	-1	0,5447	-1	0,7354
0	1,045	0	0,9003	0	0,7555	0	0,6259	0	0,4832
1	1,164	1	1,009	1	0,8578	1	0,7109	1	0,5615
2	1,298	2	1,124	2	0,9669	2	0,8107	2	0,7539
3	1,441	3	1,259	3	1,099	3	0,9985	3	1,353
4	1,6	4	1,419	4	1,294	4	1,542	4	2,649
5	1,782	5	1,628	5	1,655	5	2,159	5	4,258
6	2,001	6	1,916	6	2,131	6	2,839	6	6,157
7	2,259	7	2,262	7	2,718	7	3,584	7	8,342
8	2,55	8	2,683	8	3,353	8	4,691	8	10,81



Gb. 4.2 Cavitation buckets diagram dengan parameter thickness ratio

4.4. Pengaruh Perubahan Camber Ratio terhadap Angle of Attack dan Koefisien Gaya Angkat

Distribusi mean line dari suatu foil NACA akan diubah dengan meninggikan dan merendahkan tinggi maksimumnya secara proporsional, yaitu tidak merubah letak tinggi maksimum. Misalnya untuk foil NACA 4412 akan menjadi NACA 3412 dan NACA 2412 bila direndahkan dan NACA 5412 dan NACA 6412 bila ditinggikan. Geometri foil dapat dibentuk dengan persamaan distribusi mean line yang sudah dijelaskan dalam subbab 4.2.

Untuk camber ratio f/c tertentu dimana termasuk perubahan thickness ratio membentuk satu set *cavitation buckets diagram*. Kemudian bila semua foil NACA yang disebutkan di paragraf di atas dimasukkan dalam perhitungan maka didapat seri *cavitation buckets diagram*, jadi setiap seri terdiri dari beberapa set *cavitation buckets diagram*.

Running program menghasilkan data yang ditampilkan dalam tabel 4.2, yaitu data untuk suatu t/c tetap dengan f/c yang diubah-ubah (gb. 4.3). Kemudian bila diplotkan semua t/c maka membentuk satu seri *cavitation buckets diagram*, ditampilkan dalam gb. 4.4, gb. 4.5 dan gb. 4.6.

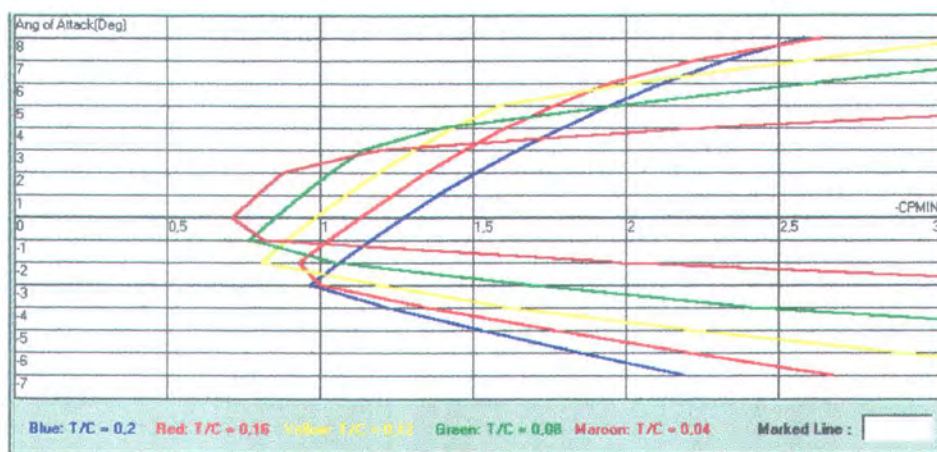
Jika hanya melihat pengaruh perubahan kelengkungan foil, camber ratio saja, data itu menghasilkan *bucket diagram-bucket diagram* yang berbeda. Perubahan itu meliputi $-C_{pmin}$ untuk kedua permukaan yaitu permukaan tekan menyempit dan permukaan hisap melebar. Perubahan yang juga terjadi adalah naiknya nilai tekanan drop maksimum saat shoc-free entry sehingga mengeser seluruh *bucket diagram* semakin dangkal. Gambar menunjukkan perubahan itu

Tabel 4-2

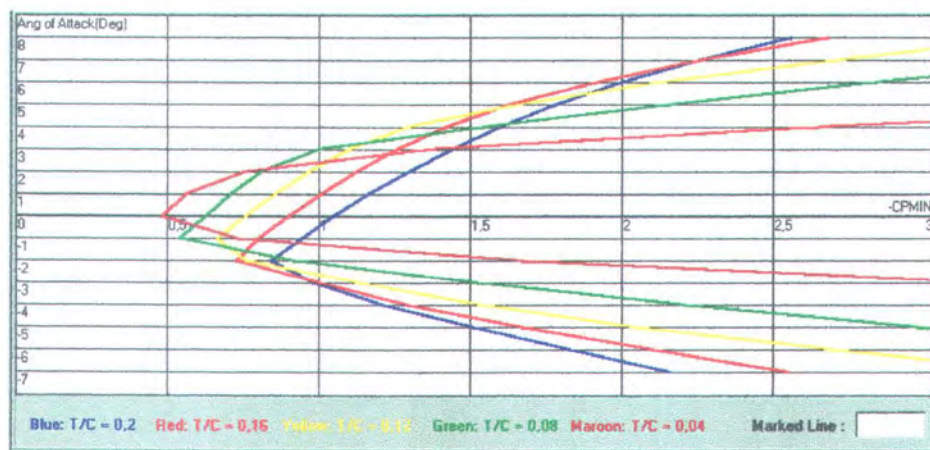
T/C= 0,12					
F/C= 0,02		F/C= 0,04		F/C= 0,06	
ALPHA	-CPMIN	ALPHA	-CPMIN	ALPHA	-CPMIN
-7	3,642	-7	3,404	-7	3,195
-6	2,894	-6	2,694	-6	2,518
-5	2,21	-5	2,044	-5	1,899
-4	1,611	-4	1,54	-4	1,447
-3	1,193	-3	1,134	-3	1,057
-2	0,8092	-2	0,7618	-2	0,7311
-1	0,8947	-1	0,6673	-1	0,4796
0	0,9847	0	0,7565	0	0,559
1	1,085	1	0,8578	1	0,6643
2	1,193	2	0,9669	2	0,8084
3	1,309	3	1,099	3	1,028
4	1,433	4	1,294	4	1,329
5	1,587	5	1,655	5	1,724
6	2,033	6	2,131	6	2,252
7	2,593	7	2,718	7	2,869
8	3,198	8	3,353	8	3,537



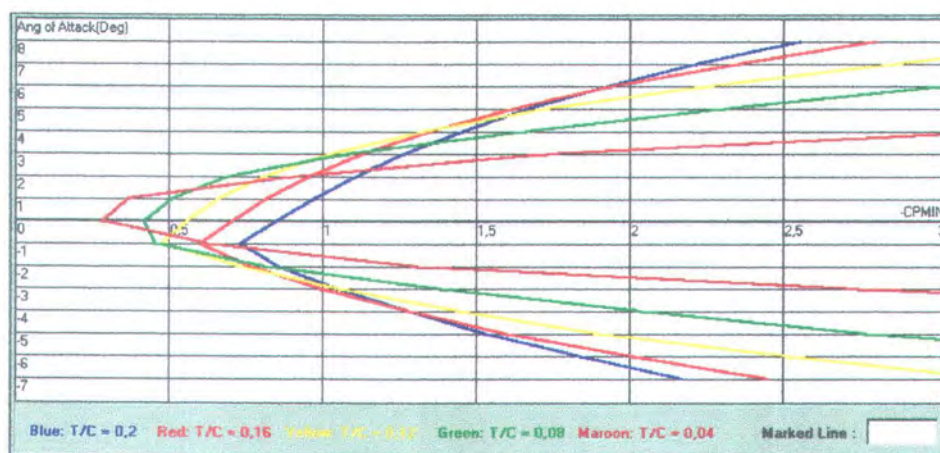
Gb. 4.3 Sebuah bucket diagram untuk camber ratio yang berbeda



Gb. 4.4 Cavitation buckets diagram untuk camber ratio 0,06



Gb. 4.5 Cavitation buckets diagram untuk camber ratio 0,04



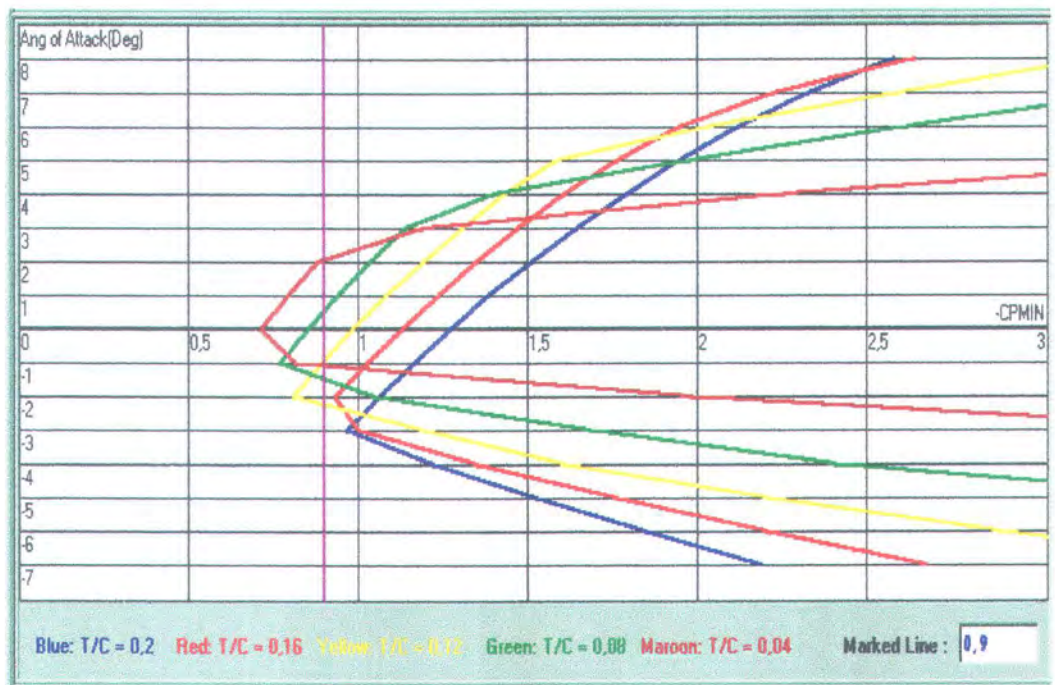
Gb. 4.6 Cavitation buckets diagram untuk camber ratio 0,02

4.5. Batas Kavitasi

Batas bebas kavitasi tergantung pada bentuk *bucket diagram* sesuai thickness ratio-nya. Batas ini diketahui dengan memplotkan angka kavitasi yang memotong semua *bucket diagram* t/c dalam *cavitation buckets diagram*, dalam gb. 4.7 angka kavitasi diambil 0,9. Dalam subbab 3.5 sudah dijelaskan bahwa awal kavitasi adalah angka kavitasi sama dengan $-C_{pmin}$, jadi penentuan batas kavitasi adalah angka kavitasi. Untuk suatu $\frac{t}{c}$, batas ini menghasilkan jangkauan α atau C_L pada angka kavitasi yang sudah ditentukan. Jangkauan ini adalah variasi dari α atau C_L yang diperbolehkan untuk mencegah terjadinya kavitasi.

$$\Delta C_L = C_{L_{hisap}} - C_{L_{tekan}}$$

$$\Delta \alpha = \alpha_{hisap} - \alpha_{tekan}$$



Gb. 4.7 Cavitation buckets diagram dan angka kavitasi

Semakin tebal suatu foil maka buckets diagram semakin lebar dan dangkal. Karakter foil tebal adalah jangkauan angle of attack atau koefisien gaya angkat besar tetapi jangkauan operasi semakin rendah. Jangkauan angle of attack besar berarti keleluasaan mendisain foil. Jangkauan operasi rendah berarti keterbatasan dalam pengoperasian foil, yang bergantung pada kondisi lingkungan, kecepatan aliran (dalam hubungannya dengan fluida viskous) sehingga rentan terhadap perubahan angka kavitasi.

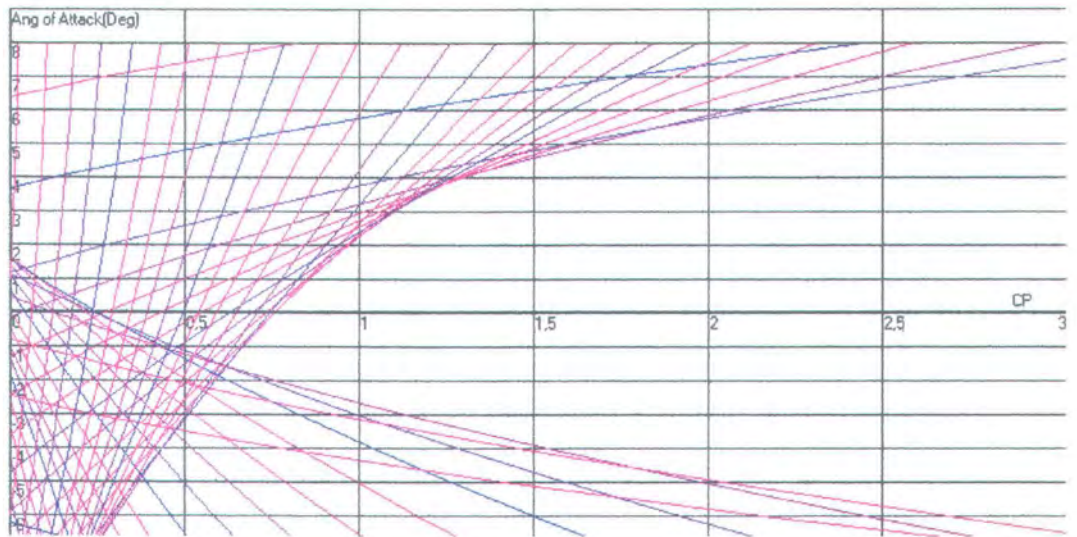
Bila angka kavitasi berada di luar bucket perlu diwaspadai terjadinya *bubble cavitation*.

Semakin cembung suatu foil maka bucket diagram semakin dangkal sedang lebar tetap asalkan $\Delta\sigma \geq \Delta\sigma_{cr}$. Lebar bucket diagram tetap karena biarpun *pressure side sheet* meningkat tetapi *back side sheet* menurun. Foil camber cenderung untuk terjadi bubble cavitation karena kenaikan nilai tekanan drop maksimum $\frac{\Delta p}{q}$ saat *shock-free entry*.

4.6. Panjang Kavitasi dan Jenis Kavitasi yang Terjadi

Cavitation buckets diagram dengan parameter $\frac{x}{c}$ dibentuk oleh gabungan grafik-grafik $-C_p$ min tiap $\frac{x}{c}$ panjang chord seperti yang sudah dijelaskan dalam subbab 3.5. Nilai-nilai distribusi tekanan yang terkecil dari setiap $\frac{x}{c}$ dari setiap angle of attack ini akan membentuk sebuah *bucket diagram*.

Hasil running program yang menunjukkan pembentukan sebuah *bucket diagram* ditampilkan dalam gb. 4.8.



Gb. 4.8 Sebuah *bucket diagram* yang dibentuk oleh $-C_{pmin}$ tiap panel

Satu *bucket diagram* menggambarkan dua bagian nilai $-C_{pmin}$, yaitu $-C_{pmin}$ untuk sisi hisap yang dimulai dari titik sudut ke atas dan $-C_{pmin}$ sisi tekan yang dimulai dari titik sudut ke bawah. Kemudian untuk $-C_{pmin}$ suatu foil diambil dari $-C_{pmin}$ tiap $\%$ tiap angle of attack sehingga bila dihubungkan membentuk suatu *bucket* (berbentuk seperti ember).

Pada sisi hisap terlihat adanya bagian yang curam, hal ini berarti di sudut-sudut rendah $-C_{pmin}$ suatu foil terjadi di daerah yang mendekati *mid-chord*. Bagian landai atas merupakan daerah *back leading edge* dimana terjadi $-C_{pmin}$ foil sedangkan bagian landai bawah adalah daerah *face leading edge* dimana terjadi $-C_{pmin}$ foil.

Untuk mendapatkan titik kavitasi, angka kavitasi diplotkan dalam grafik dan ordinat yang diinginkan sesuai dengan besarnya angle of attack. Posisi titik kavitasi dapat dicari bila grafik $\%$ semakin rapat. Panjang kavitasi dapat dicari

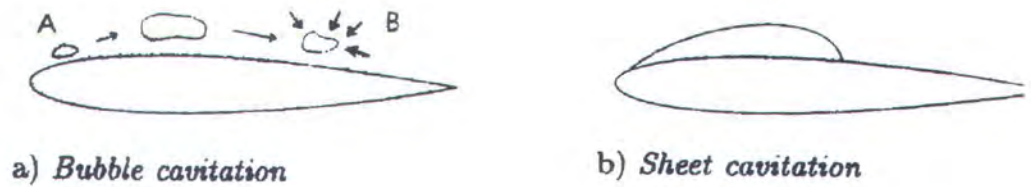
dengan menghitung selisih posisi titik kavitasi dan posisi dimana terjadi batas $-C_{p\min}$ foil pada angle of attack tersebut.

Jenis kavitasi yang terjadi dapat diterangkan bila mempertimbangkan karakter *boundary layer* dan *nuclei content*. Hal ini sudah diluar batasan dari tugas akhir karena *boundary layer* terjadi dalam fluida sejati. Dalam subbab ini hanya dibahas sampai dimana kontribusi *cavitation buckets diagram* terhadap jenis kavitasi yang terjadi.

Jenis kavitasi yang terjadi sangat dipengaruhi oleh daerah kavitasi, *nuclei content*, dan *boundary layer*. Bila panjang kavitasi berada di daerah *back mid-chord* dan terjadi *normal boundary layer transition* maka akan terbentuk *back bubble cavitation*, biasanya terjadi pada foil tebal dengan angle of attack kecil. Bila panjang kavitasi berada di daerah *back leading edge* atau *face leading edge* dan terjadi *laminar separation bubble* maka akan terbentuk *back sheet cavitation* atau *face sheet cavitation*, biasanya terjadi pada foil tipis dengan angle of attack besar.

Bubbel cavitation terjadi ketika ada tekanan rendah di daerah (A), tanpa kemiringan grafik koefisien tekanan yang curam. Kemudian *nuclei* di dalam fluida mengembang cepat, dan, bergerak seiring aliran, *nuclei* ini akan pecah di daerah (B) dimana daerah tekanan uap terlampaui, ditampilkan dalam gb. 4.9.

Sheet cavitation muncul di daerah yang berbeda dengan kemiringan grafik koefisien tekanan yang curam. Kondisi ini berada di *nose leading edge*, biasanya foil tipis dengan angle of attack yang lebih besar, ditampilkan dalam gb. 4.9.



Gb. 4.9 Jenis kavitasi yang terjadi

4.7. Geometri Foil yang Optimum

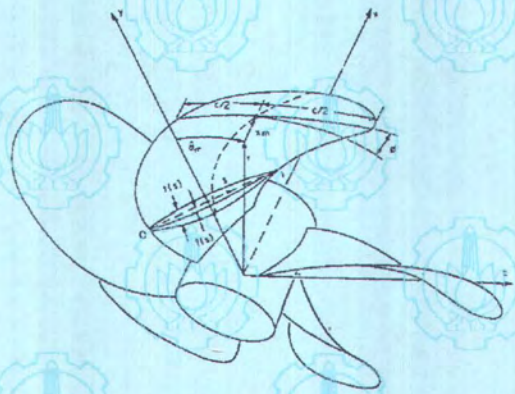
Kedua cavitation buckets diagram yaitu $-C_{pmin}$ lawan α dan $-C_{pmin}$ lawan C_L mempunyai analisa yang sama, jadi salah satu diagram akan dijadikan sebagai basis analisa. Efisiensi foil akan lebih jelas jika C_L dijadikan ukuran karena hal ini adalah salah satu kompromi disain antara kavitasi dan gaya angkat dalam suatu disain utuh.

Dalam satu bucket diagram ditunjukkan peningkatan koefisien gaya angkat. Peningkatan yang berarti ditunjukkan dengan adanya tekanan rendah di permukaan atas foil, $-C_{pmin}$ bertambah. Koefisien meningkat cepat setelah titik batas antara *face cavitation* dan *back cavitation* terlampaui, grafik naik tajam. Kemudian penambahan koefisien menurun tetapi masih mempunyai slope yang positif walau kecil, bentuk grafik naik landai.

Meninjau hanya satu *bucket diagram* koefisien akan bernilai tinggi tercapai jika $-C_{pmin}$ yang besar tetapi hal ini tidak efisien karena penambahan sedikit koefisien gaya angkat menghasilkan angka kavitasi yang besar terutama untuk foil-foil tipis. Efisiensi tercapai dimulai dari titik batas koefisien antara *face cavitation* dan *back cavitation* sampai peningkatan koefisien mulai menurun. Jadi

setiap foil dengan γ_c tertentu mempunyai nilai optimum yaitu saat grafik berubah dari bentuk curam ke bentuk landai.

Untuk mengetahui nilai optimum suatu geometri foil dapat dianalisa dengan *cavitation buckets diagram* dengan parameter camber ratio. Nilai optimum dapat diperoleh dengan sembaran γ_c dan sembarang $-C_{pmin}$ untuk setiap camber ratio γ_c tertentu. Hal ini dapat diperoleh dengan membuat bucket diagram dengan parameter γ_c sebanyak mungkin sehingga grafik satu dengan yang lainnya semakin rapat. Nilai optimum adalah nilai koefisien tertinggi pada sembarang γ_c untuk setiap $-C_{pmin}$ -nya. Kemudian grafik geometri optimum dibuat dengan $-C_{pmin}$ lawan γ_c dengan parameter γ_c .



BAB V

KESIMPULAN DAN SARAN

BAB V

KESIMPULAN DAN SARAN



5.1. Kesimpulan

Dari pembahasan pada bab-bab sebelumnya, maka pengaruh perubahan thickness ratio dan camber ratio suatu foil agar terhindar dari kavitasi dapat disimpulkan :

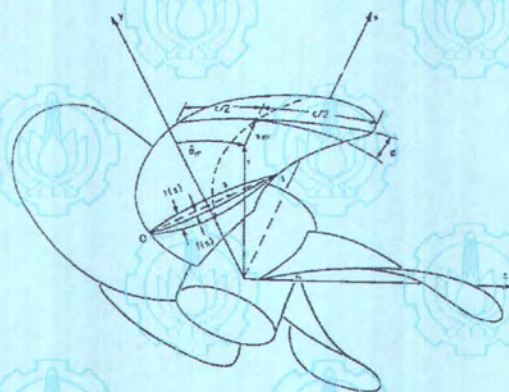
1. *Cavitation buckets diagram* memberikan keleluasaan disain foil karena tersedia pilihan-pilihan thickness ratio, camber ratio, dan angle of attack.
2. Karakter kavitasi banyak dipengaruhi oleh :
 - Tebal, yaitu thickness ratio, dan distribusi tebal.
 - Camber, yaitu camber ratio, dan distribusi mean line.
3. Bila thickness ratio meningkat :
 - Memperbesar variasi angle of attack dan ΔC_L .
 - Memperbesar kemungkinan terjadinya *back bubble cavitation* dan memperkecil *sheet cavitation* karena adanya pergeseran letak *shock-free entry*.
4. Bila camber ratio meningkat :
 - Variasi angle of attack dan ΔC_L tidak banyak berubah. *Suction sheet cavitation* memang berkurang tetapi *pressure sheet cavitation* bertambah,

sehingga menggeser titik batas angle of attack antara *face cavitation* dan *back cavitation* ke atas, angle of attack bertambah.

- Seluruh *bucket diagram* akan bergeser ke kanan sehingga akan memperbesar kemungkinan terjadinya *back bubble cavitation*.
4. Koefisien gaya angkat yang besar dapat dicapai saat kondisi *shock-free entry*.
 5. Sheet cavitation terjadi pada face leading-edge dan back leading-edge sedangkan back bubble cavitation terjadi pada daerah mid-chord atau tebal maksimum.
 6. Panjang kavitasi dihitung dari posisi angka kavitasi sampai bucket diagram setiap angle of attack-nya. Posisi kavitasi dapat diketahui dengan menghitung puncak koefisien tekanan minimum ($-C_{pmin}$) setiap $\frac{1}{2}$ panjang foil.
 7. Geometri foil yang optimum adalah geometri yang memiliki koefisien gaya angkat sebesar mungkin dan angka kavitasi sekecil mungkin untuk setiap camber ration tertentu.

5.2. Saran

1. Penggunaan lebih banyak panel untuk menghitung distribusi tekanan permukaan hydrofoil akan menampilkan data yang lebih akurat sehingga dapat meningkatkan hasil analisa kavitasi.
2. Pengkajian karakter boundary layer dapat menerangkan terjadinya jenis-jenis kavitasi bila letak atau panjang kavitasi diketahui.



DAFTAR PUSTAKA

DAFTAR PUSTAKA

Abbott, Ira H. and Doenhoff, Albert E. von, "*Theory of Wing Section, Including a Summary of Airfoil Data*", Dover Publication, Inc., New York, 1958.

Boswell, Robert J. and Cox, Geoffrey G., "*Design and Model Evaluation of a Highly Skewed Propeller for a Cargo Ship*", The Society of Naval Architects and Marine Engineers, New York Section, 1974.

Gunsteren, L.A. van and Pronk, C., "*Propeller Design Concepts*", The Society of Naval Architects and Marine Engineers, New York Section, 1977.

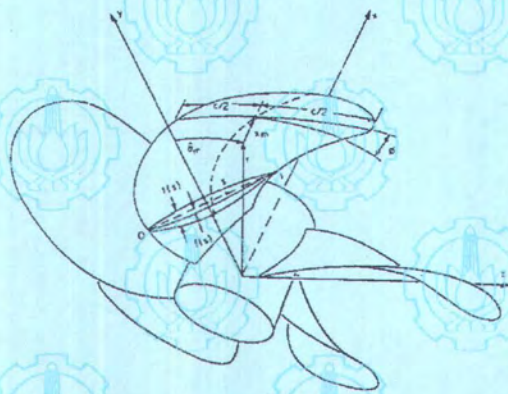
Holtrop, J., "*Propeller Design*", Maritime Research Institute Netherlands (MARIN), 1992.

Kerwin, Justin E., "*Hydrofoil dan Propeller*", Lectures Notes, Massachusetts Institute of Technology, Massachusetts, 1993.

Kuethe, Arnold M. and Chow, Chuen-Yen, "*Foundation of Aerodynamics*", Fourth Edition, John Wiley & Sons Inc., 1986.

Lewis, Edward V., Editor, "*Principles of Naval Architecture, Volume II. Resistance, Propulsion and Vibration*", Second Revision, The Society of Naval Architects and Marine Engineers, 601 Pavonia Avenue, Jersey City, 1988.

Olson, Reuben M. and Wrigth, Steven J., "*Essentials of Engineering Fluid Mechanics*", Fifth Edition, Harper & Row Publishers Inc., 1990.



LAMPIRAN

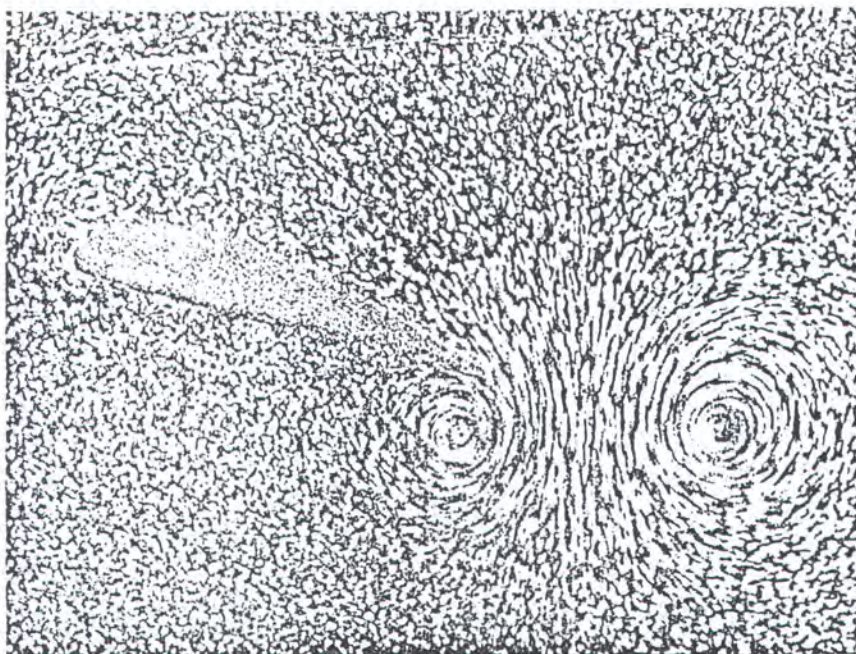


Fig. 12. Particle paths after sudden stop of airfoil in Fig. 11.

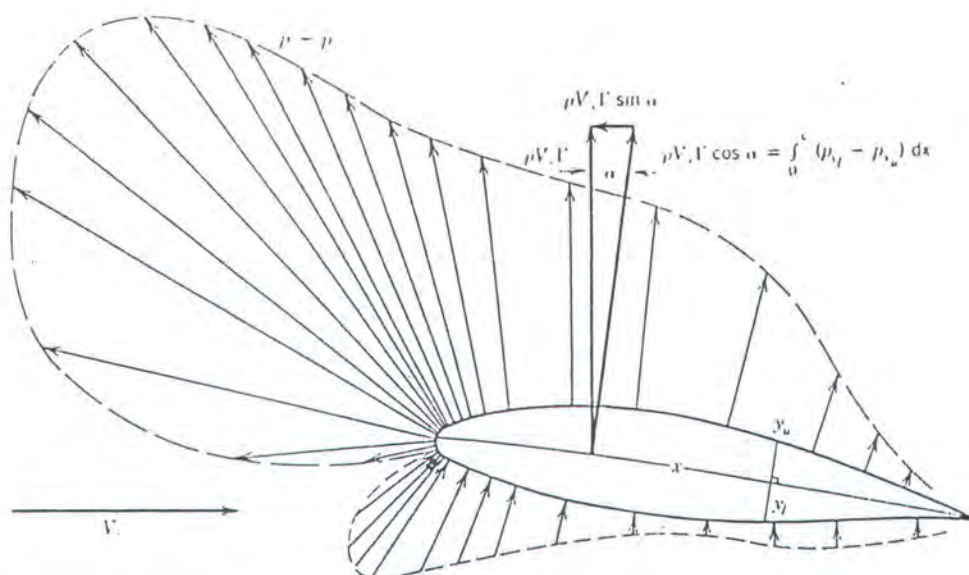
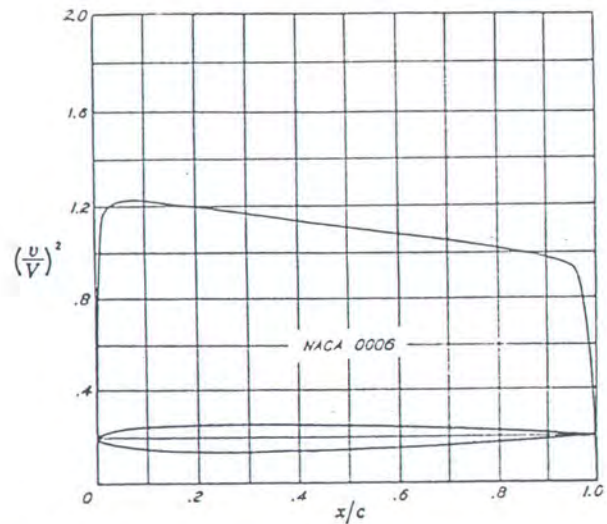


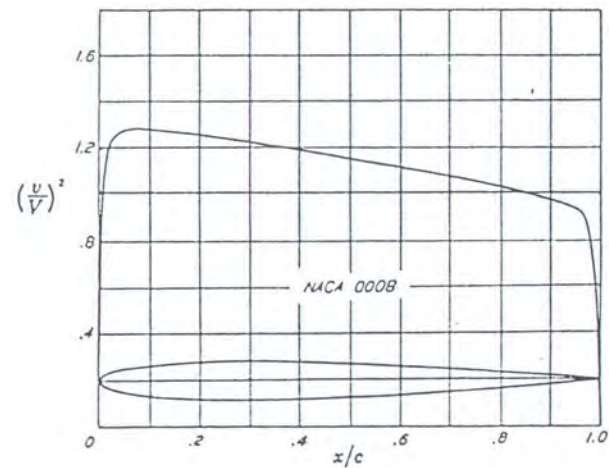
Fig. 13. Calculated pressure distribution on NACA 65₃ - 418 airfoil at $\alpha = 8^\circ$. (Courtesy of Airflow Sciences Corp.)



x (per cent c)	y (per cent c)	$(v/V)^2$	r/V	$\Delta r_a/V$
0	0	0	0	3.992
0.5	0.880	0.938	2.015
1.25	0.047	1.117	1.057	1.364
2.5	1.307	1.186	1.089	0.984
5.0	1.777	1.217	1.103	0.696
7.5	2.100	1.225	1.107	0.562
10	2.341	1.212	1.101	0.478
15	2.673	1.206	1.098	0.378
20	2.869	1.190	1.091	0.316
25	2.971	1.179	1.086	0.272
30	3.001	1.162	1.078	0.239
40	2.902	1.136	1.066	0.189
50	2.647	1.109	1.053	0.152
60	2.282	1.086	1.042	0.123
70	1.832	1.057	1.028	0.097
80	1.312	1.026	1.013	0.073
90	0.724	0.980	0.990	0.047
95	0.403	0.949	0.974	0.032
100	0.063	0	0	0

L.E. radius: 0.40 per cent c

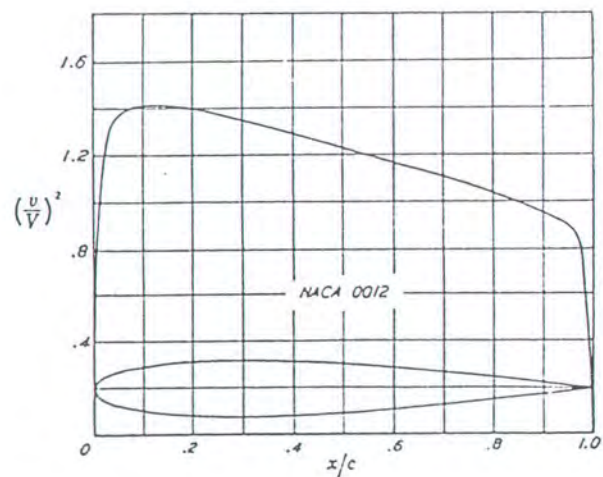
NACA 0006 Basic Thickness Form



x (per cent c)	y (per cent c)	$(v/V)^2$	r/V	$\Delta r_a/V$
0	0	0	0	2.900
0.5	0.792	0.890	1.795
1.25	1.263	1.103	1.050	1.310
2.5	1.743	1.221	1.105	0.971
5.0	2.360	1.272	1.128	0.694
7.5	2.800	1.284	1.133	0.561
10	3.121	1.277	1.130	0.479
15	3.564	1.272	1.128	0.379
20	3.825	1.259	1.122	0.318
25	3.961	1.241	1.114	0.273
30	4.001	1.223	1.106	0.239
40	3.869	1.186	1.089	0.188
50	3.529	1.149	1.072	0.152
60	3.043	1.111	1.054	0.121
70	2.443	1.080	1.039	0.096
80	1.749	1.034	1.017	0.071
90	0.965	0.968	0.984	0.047
95	0.537	0.939	0.969	0.031
100	0.084	0

L.E. radius: 0.70 per cent c

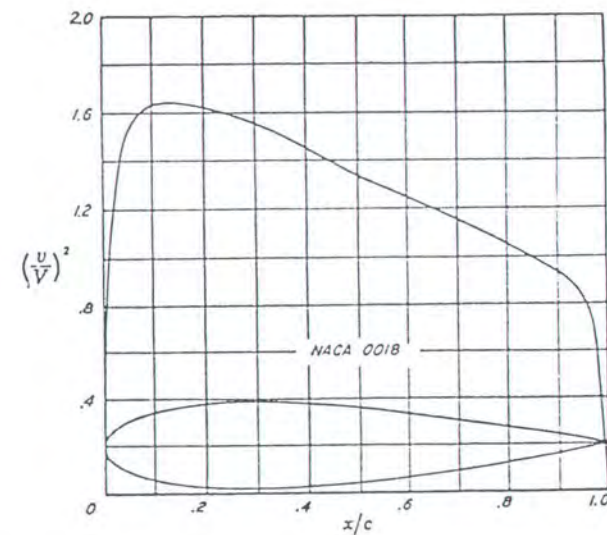
NACA 0008 Basic Thickness Form



x (per cent c)	y (per cent c)	$(v/V)^2$	v/V	$\Delta v_a/V$
0	0	0	0	1.988
0.5	0.640	0.800	1.475
1.25	1.894	1.010	1.005	1.199
2.5	2.615	1.241	1.114	0.934
5.0	3.555	1.378	1.174	0.685
7.5	4.200	1.402	1.184	0.558
10	4.683	1.411	1.188	0.479
15	5.345	1.411	1.188	0.381
20	5.737	1.399	1.183	0.319
25	5.941	1.378	1.174	0.273
30	6.002	1.350	1.162	0.239
40	5.803	1.288	1.135	0.187
50	5.294	1.228	1.108	0.149
60	4.563	1.166	1.080	0.118
70	3.664	1.109	1.053	0.092
80	2.623	1.044	1.022	0.068
90	1.448	0.956	0.978	0.044
95	0.807	0.906	0.952	0.029
100	0.126	0	0	0

L.E. radius: 1.58 per cent c

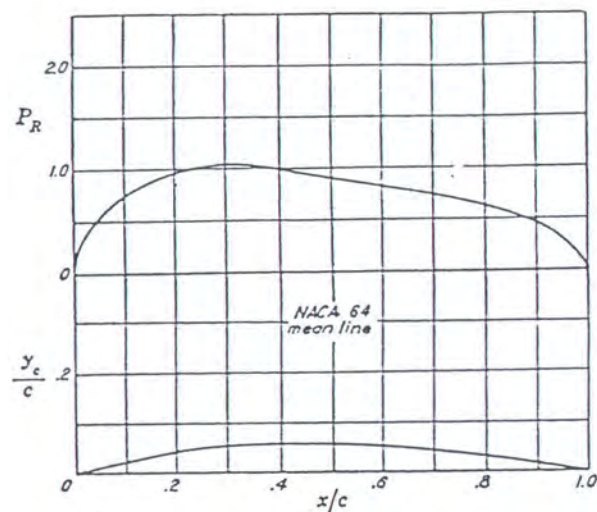
NACA 0012 Basic Thickness Form



x (per cent c)	y (per cent c)	$(v/V)^2$	v/V	$\Delta v_a/V$
0	0	0	0	1.342
0.5	0.465	0.682	1.178
1.25	2.841	0.857	0.926	1.028
2.5	3.922	1.217	1.103	0.861
5.0	5.332	1.507	1.228	0.662
7.5	6.300	1.598	1.264	0.555
10	7.024	1.628	1.276	0.479
15	8.018	1.633	1.278	0.381
20	8.606	1.625	1.275	0.320
25	8.912	1.592	1.262	0.274
30	9.003	1.556	1.247	0.238
40	8.705	1.453	1.205	0.184
50	7.941	1.331	1.154	0.144
60	6.845	1.246	1.116	0.113
70	5.496	1.153	1.074	0.087
80	3.935	1.051	1.025	0.063
90	2.172	0.933	0.966	0.039
95	1.210	0.836	0.914	0.025
100	0.189	0	0	0

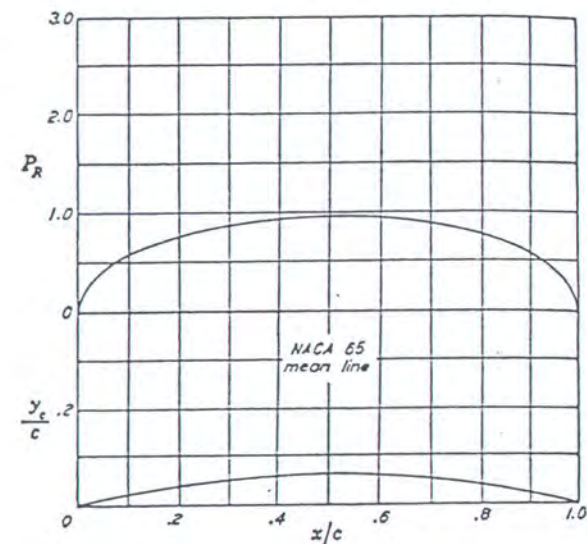
L.E. radius: 3.56 per cent c

NACA 0018 Basic Thickness Form



$c_{l_i} = 0.76 \quad \alpha_i = 0.74^\circ \quad c_{m_{c/4}} = -0.157$				
x (per cent c)	y_c (per cent c)	dy_c/dx	P_R	$\Delta v/V = P_R/4$
0	0	0.30000	0	0
1.25	0.369	0.29062	0.257	0.064
2.5	0.726	0.28125	0.391	0.098
5.0	1.406	0.26250	0.546	0.137
7.5	2.039	0.24375	0.668	0.167
10	2.625	0.22500	0.748	0.187
15	3.656	0.18750	0.871	0.218
20	4.500	0.15000	0.966	0.242
25	5.156	0.11250	1.030	0.258
30	5.625	0.07500	1.040	0.260
40	6.000	0	0.999	0.250
50	5.833	-0.03333	0.910	0.228
60	5.333	-0.06667	0.827	0.207
70	4.500	-0.10000	0.750	0.188
80	3.333	-0.13333	0.635	0.159
90	1.833	-0.16667	0.466	0.117
95	0.958	-0.18333	0.334	0.084
100	0	-0.20000	0	0

Data for NACA Mean Line 64



$c_{l_i} = 0.75 \quad \alpha_i = 0^\circ \quad c_{m_{c/4}} = -0.187$				
x (per cent c)	y_c (per cent c)	dy_c/dx	P_R	$\Delta v/V = P_R/4$
0	0	0.24000	0	0
1.25	0.296	0.23400	0.205	0.051
2.5	0.585	0.22800	0.294	0.074
5.0	1.140	0.21600	0.413	0.103
7.5	1.665	0.20400	0.502	0.126
10	2.160	0.19200	0.571	0.143
15	3.060	0.16800	0.679	0.170
20	3.840	0.14400	0.760	0.190
25	4.500	0.12000	0.824	0.206
30	5.040	0.09600	0.872	0.218
40	5.760	0.04800	0.932	0.233
50	6.000	0	0.951	0.238
60	5.760	-0.04800	0.932	0.233
70	5.040	-0.09600	0.872	0.218
80	3.840	-0.14400	0.760	0.190
90	2.160	-0.19200	0.571	0.143
95	1.140	-0.21600	0.413	0.103
100	0	-0.24000	0	0

Data for NACA Mean Line 65

NACA 4412

(Stations and ordinates given in
per cent of airfoil chord)

Upper surface		Lower surface	
Station	Ordinate	Station	Ordinate
0	0	0	0
1.25	2.44	1.25	- 1.43
2.5	3.39	2.5	- 1.95
5.0	4.73	5.0	- 2.49
7.5	5.76	7.5	- 2.74
10	6.59	10	- 2.86
15	7.89	15	- 2.88
20	8.80	20	- 2.74
25	9.41	25	- 2.50
30	9.76	30	- 2.26
40	9.80	40	- 1.80
50	9.19	50	- 1.40
60	8.14	60	- 1.00
70	6.69	70	- 0.65
80	4.89	80	- 0.39
90	2.71	90	- 0.22
95	1.47	95	- 0.16
100	(0.13)	100	(- 0.13)
100	100	0
L.E. radius: 1.58			
Slope of radius through L.E.: 0.20			

NACA 4415

(Stations and ordinates given in
per cent of airfoil chord)

Upper surface		Lower surface	
Station	Ordinate	Station	Ordinate
0	0	0
1.25	3.07	1.25	- 1.79
2.5	4.17	2.5	- 2.48
5.0	5.74	5.0	- 3.27
7.5	6.91	7.5	- 3.71
10	7.84	10	- 3.98
15	9.27	15	- 4.18
20	10.25	20	- 4.15
25	10.92	25	- 3.98
30	11.25	30	- 3.75
40	11.25	40	- 3.25
50	10.53	50	- 2.72
60	9.30	60	- 2.14
70	7.63	70	- 1.55
80	5.55	80	- 1.03
90	3.08	90	- 0.57
95	1.67	95	- 0.36
100	(0.16)	100	(- 0.16)
100	100	0
L.E. radius: 2.48			
Slope of radius through L.E.: 0.20			

NACA 4418

(Stations and ordinates given in
per cent of airfoil chord)

Upper surface		Lower surface	
Station	Ordinate	Station	Ordinate
0	0	0
1.25	3.76	1.25	- 2.11
2.5	5.00	2.5	- 2.99
5.0	6.75	5.0	- 4.06
7.5	8.06	7.5	- 4.67
10	9.11	10	- 5.06
15	10.66	15	- 5.49
20	11.72	20	- 5.56
25	12.40	25	- 5.49
30	12.76	30	- 5.26
40	12.70	40	- 4.70
50	11.85	50	- 4.02
60	10.44	60	- 3.24
70	8.55	70	- 2.45
80	6.22	80	- 1.67
90	3.46	90	- 0.93
95	1.89	95	- 0.55
100	(0.19)	100	(- 0.19)
100	100	0
L.E. radius: 3.56			
Slope of radius through L.E.: 0.20			

```

unit NACA;
interface
uses Foil,MeanLines;
Type
  TNACA=Class
    SERIES:String;
    PANELCOUNT:integer;
    MLine:Array[0..2]of TMeanLines;
  Constructor
NACA(SERIAL:String;m:integer);
    end;
implementation

Constructor
TNACA.NACA(SERIAL:String;m:integer);
var
  i:integer;
  STEP, YM,XFC,FC:Double;
begin
  SERIES:=Copy(SERIAL,1,LENGTH(SERIAL));
  PANELCOUNT:=m;
  Val(Copy(SERIES,2,1),XFC,i);
  XFC:=XFC/10;
  Val(Copy(SERIES,3,2),YM,i);
  YM:=YM/100;
  Val(Copy(SERIES,1,1),FC,i);
  STEP:=0.5*FC/100;
  FC:=FC/100+2*STEP;
  for i:=0 to 2 do
  begin
    FC:=FC-STEP;
    MLine[i]:=TMeanLines.MeanLines(PANELCOUNT,FC,XFC,YM);
  end;
end;

end.

```

```

unit MeanLines;

interface
uses Foil;
  Type
  TMeanLines=Class
  PANELCOUNT:integer;
  FC:Double;
  PFoil:Array[0..4]Of TFoil;
  Constructor
Meanlines(m:integer;TFC, TXFC, TTC:Double)
;
  end;
implementation

Constructor
TMeanlines.Meanlines(m:integer;TFC, TXFC,
TTC:Double);
var
  i:integer;
  STEP,TC:Double;
begin
PANELCOUNT:=m;
FC:=TFC;
TC:=TTC;
STEP:=TC/3;
TC:=TC+3*STEP;
for i:=0 to 4 do
begin
TC:=TC-STEP;
PFoil[i]:=TFoil.Foil(PANELCOUNT,FC, TXF
C,TC);
end;
end;

end.

```



```

unit Foil;

interface
uses SysUtils, Math, Panels;
Type
  TFoil=Class
  public
    TC, LERAD: Double;
    XS, YS: Array[0..10000] Of Double;
    XB, YB: Variant;
    FPanels: Array[0..15] Of TPanels;
  Constructor
Foil(m: integer; TFC, TXFC, TTC: Double);
  procedure SetPanel;
  private
    PANELCOUNT: integer;
    FC, XFC: Double;
    XT, YC, YT, THETAC: Array[0..5000] Of
Double;
  procedure SetPanelBoundary;
  procedure SetFoil;
  Function FindPoint(p: Double): integer;
  end;

implementation
Constructor
TFoil.Foil(m: integer; TFC, TXFC, TTC: Double)
;
begin
  PANELCOUNT:=m;
  FC:=TFC;
  XFC:=TXFC;
  TC:=TTC;
  SetFoil;
  SetPanelBoundary;
end;

procedure TFoil.SetFoil;
var
  i: integer;
begin
  for i:=0 to High(XT) do
  begin
    XT[i]:=i/High(XT);
    YT[i]:=(TC*(1.4845*sqrt(XT[i])-0.63*XT[i]-
1.758*sqrt(XT[i])+1.4215*Power(XT[i],3)-
0.5075*Power(XT[i],4)));
  end;
  for i:=0 to High(XT) do
  if(XT[i]<=XFC)Then
  begin
    YC[i]:=(FC/SQR(XFC))*(2*XFC*XT[i]-
SQR(XT[i]));
    THETAC[i]:=ArcTan((FC/SQR(XFC))*2*(XFC-XT[i]));
  end
  else
  begin
    YC[i]:=(FC/SQR(1-XFC))*(1-
2*XFC+2*XFC*XT[i]-SQR(XT[i]));

```

```

    THETAC[i]:=ArcTan((FC/SQR(1-
XFC))*2*(XFC-XT[i]));
  end;
  for i:=0 to High(XT) do
  begin
    XS[i]:=XT[High(XT)-i]+YT[High(XT)-
i]*Sin(THETAC[High(XT)-i]);
    YS[i]:=YC[High(XT)-i]-YT[High(XT)-
i]*Cos(THETAC[High(XT)-i]);
  end;
  for i:=1 to High(XT) do
  begin
    XS[High(XT)+i]:=XT[i]-
YT[i]*Sin(THETAC[i]);
    YS[High(XT)+i]:=YC[i]+YT[i]*Cos(THETA
C[i]);
  end;
  XS[0]:=1;
  XS[High(XS)]:=1;
  YS[0]:=0;
  YS[High(XS)]:=0;
end;

procedure TFoil.SetPanelBoundary;
var
  i, j: integer;
  ANGLE: Double;
begin
  XB:=VarArrayCreate([1, PANELCOUNT+1], v
arDouble);
  YB:=VarArrayCreate([1, PANELCOUNT+1], v
arDouble);
  for i:=0 to PANELCOUNT do
  begin
    ANGLE:=(i/PANELCOUNT)*360;
    j:=FindPoint(ANGLE);
    XB[i+1]:=XS[j];
    YB[i+1]:=YS[j];
  end;
end;

procedure TFoil.SetPanel;
var
  i, j: integer;
begin
  j:=-8;
  for i:= 0 to 15 do
  begin
    inc(j);
    FPanels[i]:=TPanels.Panels(j, XB, YB);
  end;
end;

Function TFoil.FindPoint(p: Double): integer;
var
  i: integer;
  Minimal, X: Double;
  REST1: Array[0..5000] of Double;
  REST2: Array[5001..10000] of Double;
begin

```

```

X:=0.5*(1+Cos(DegToRad(p)));
if(p<=180)Then
begin
for i:=0 to High(REST1) do
REST1[i]:=abs(XS[i]-X);
Minimal:=MinValue(REST1);
for i:=0 to High(REST1) do
if(REST1[i]=Minimal)Then
begin
Result:=i;
break;
end;
end
else
begin
for i:=5001 to High(REST2) do
REST2[i]:=abs(XS[i]-X);
Minimal:=MinValue(REST2);
for i:=5001 to High(REST2) do
if(REST2[i]=Minimal)Then
begin
Result:=i;
break;
end;
end;
end;

end.

```

unit Panels;

interface

Uses Math, Matrix, Variants;

Type

TPanel = Class

Attack: Double;

SumPanel: integer;

XB, YB, X, Y, S, THETA, GAMMA, VP, CP, CL:

Variant;

CPMIN: Double;

Constructor Panels(r: Double; X, Y: Variant);

private

procedure SetPanelParameter;

procedure SetDistribution;

end;

implementation

Constructor

TPanel.Panels(r: Double; X, Y: Variant);

begin

Attack := DegToRad(r);

varCopy(XB, X);

varCopy(YB, Y);

SumPanel := varArrayHighBound(XB, 1) - 1;

SetDistribution;

end;

procedure TPanel.SetPanelParameter;

var

i, PL1: integer;

begin

X := VarArrayCreate([1, SumPanel], varDouble);

Y := VarArrayCreate([1, SumPanel], varDouble);

THETA := VarArrayCreate([1, SumPanel], varDouble);

S := VarArrayCreate([1, SumPanel], varDouble);

for i := 1 to SumPanel do

begin

PL1 := i + 1;

X[i] := 0.5 * (XB[i] + XB[PL1]);

Y[i] := 0.5 * (YB[i] + YB[PL1]);

THETA[i] := ARCTAN2((YB[PL1] -

YB[i]), (XB[PL1] - XB[i]));

S[i] := sqrt(SQR(XB[PL1] -

XB[i]) + SQR(YB[PL1] - YB[i]));

end;

end;

procedure TPanel.SetDistribution;

var

i, j, k: integer;

A, B, C, D, E, F, G, P, Q: Double;

AN, AT, CN1, CN2, CT1, CT2, SINE, COSINE, R

HS, TEMP: Variant;

begin

SetPanelParameter;

SINE := VarArrayCreate([1, SumPanel], varDouble);

COSINE := VarArrayCreate([1, SumPanel], varDouble);

for i := 1 to SumPanel do

begin

SINE[i] := SIN(THETA[i]);

COSINE[i] := COS(THETA[i]);

end;

RHS := VarArrayCreate([1, SumPanel + 1], varDouble);

for i := 1 to SumPanel do

RHS[i] := SIN(THETA[i] - Attack);

RHS[SumPanel + 1] := 0;

GAMMA := VarArrayCreate([1, SumPanel + 1], varDouble);

VP := VarArrayCreate([1, SumPanel], varDouble);

CP := VarArrayCreate([1, SumPanel], varDouble);

CL := VarArrayCreate([1, SumPanel], varDouble);

TEMP := VarArrayCreate([1, SumPanel, 1, 2], varDouble);

CN1 := VarArrayCreate([1, SumPanel, 1, SumPanel], varDouble);

CN2 := VarArrayCreate([1, SumPanel, 1, SumPanel], varDouble);

CT1 := VarArrayCreate([1, SumPanel, 1, SumPanel], varDouble);

CT2 := VarArrayCreate([1, SumPanel, 1, SumPanel], varDouble);

AN := VarArrayCreate([1, SumPanel + 1, 1, SumPanel + 1], varDouble);

AT := VarArrayCreate([1, SumPanel, 1, SumPanel + 1], varDouble);

for i := 1 to SumPanel do

for j := 1 to SumPanel do

if (i = j) then

begin

CN1[i, j] := -1;


```

    CN2[i,j]:=1;
    CT1[i,j]:=0.5*PI;
    CT2[i,j]:=0.5*PI;
end
else
begin
    A:=(XB[j]-X[i])*COSINE[j]-(Y[i]-
YB[j])*SINE[j];
    B:=SQR(X[i]-XB[j])+SQR(Y[i]-YB[j]);
    C:=Sin(THETA[i]-THETA[j]);
    D:=Cos(THETA[i]-THETA[j]);
    E:=((X[i]-XB[j])*SINE[j]-(Y[i]-
YB[j])*COSINE[j]);
    F:=Ln(1+((SQR(S[j])+2*A*S[j])/B));
    G:=ArcTan2(E*S[j],B+A*S[j]);
    P:=(X[i]-XB[j])*SIN(THETA[i]-
2*THETA[j])+(Y[i]-YB[j])*COS(THETA[i]-
2*THETA[j]);
    Q:=(X[i]-XB[j])*COS(THETA[i]-
2*THETA[j])-(Y[i]-YB[j])*SIN(THETA[i]-
2*THETA[j]);
    CN2[i,j]:=D+0.5*Q*F/S[j]-
(A*C+D*E)*G/S[j];
    CN1[i,j]:=0.5*D*F+C*G-CN2[i,j];
    CT2[i,j]:=C+0.5*P*F/S[j]+(A*D-
C*E)*G/S[j];
    CT1[i,j]:=0.5*C*F-D*G-CT2[i,j];
end;
for i:=1 to SumPanel do
begin
    AN[i,1]:=CN1[i,1];
    AN[i,SumPanel+1]:=CN2[i,SumPanel];
    AT[i,1]:=CT1[i,1];
    AT[i,SumPanel+1]:=CT2[i,SumPanel];
for j:=2 to SumPanel do
begin
    AN[i,j]:=CN1[i,j]+CN2[i,j-1];
    AT[i,j]:=CT1[i,j]+CT2[i,j-1];
end;
end;
AN[SumPanel+1,1]:=1;
AN[SumPanel+1,SumPanel+1]:=1;
for j:=2 to SumPanel do
    AN[SumPanel+1,j]:=0;
    GAMMA:=Gaussian(AN,RHS);
    for i:=1 to SumPanel do
    begin
        VP[i]:=Cos(THETA[i]-Attack);
        for j:=1 to SumPanel+1 do
        begin
            VP[i]:=VP[i]+AT[i,j]*GAMMA[j];
            CP[i]:=1-SQR(VP[i]);
        end;
        TEMP[i,1]:=CP[i];
        CL[i]:=CP[i]*Cos(THETA[i]);
    end;
    j:=FindIndexMin(1,TEMP);
    CPMIN:=CP[j];
end;
end;
end.

```

```

unit BSpline;

interface
uses Math, Matrix, Mathematics, Variants;
Type
  TBSpline=class
  public
    nData,nSegment:integer;
    BSP: Variant;
    Constructor
BSpline(DTS: Variant;nknot:integer);
    Constructor
OBSpline(DTS: Variant;nknot:integer);
  private
    knot:integer;
    Function SetN: Variant;
    Function SetT(p:Double): Variant;
    Function
SetB(o:integer;DTS: Variant): Variant;
    Function
SetOB(o:integer;DTS: Variant): Variant;
  end;

implementation

Constructor
TBSpline.BSpline(DTS: Variant;nknot:integer)
;
var
  i,j,k:integer;
  TEMPORER: Variant;
begin
  knot:=nknot;
  nSegment:=varArrayHighBound(DTS,1);
  nData:=100*nSegment;

BSP:=varArrayCreate([1,nData,1,2],varDouble
);
  k:=0;
  for i:=0 to nSegment-1 do
  for j:=0 to 99 do
  begin
    inc(k);

varCopy(TEMPORER,MulMatrix(MulMatrix(
SetT(j*0.01),SetN),SetB(i,DTS)));
    BSP[k,1]:=TEMPORER[1];
    BSP[k,2]:=TEMPORER[2];
  end;
end;

Constructor
TBSpline.OBSpline(DTS: Variant;nknot:integer);
var
  i,j,k:integer;
  TEMPORER: Variant;
begin
  knot:=nknot;
  nSegment:=varArrayHighBound(DTS,1)-1;

```

```

  nData:=100* nSegment;

BSP:=varArrayCreate([1,nData,1,2],varDouble
);
  k:=0;
  for i:=0 to nSegment-1 do
  for j:=0 to 99 do
  begin
    k:=k+1;

varCopy(TEMPORER,MulMatrix(MulMatrix(
SetT(j*0.01),SetN),SetB(i,DTS)));
    BSP[k,1]:=TEMPORER[1];
    BSP[k,2]:=TEMPORER[2];
  end;
end;

Function TBSpline.SetT(p:Double): Variant;
var
  i:integer;
begin

Result:=VarArrayCreate([1,knot],varDouble);
  for i:=1 to knot do
    Result[i]:=Power(p,knot-i);
  end;

Function TBSpline.SetN: Variant;
var
  i,j,l:integer;
  M,N:Double;
begin

Result:=varArrayCreate([1,knot,1,knot],varDouble);
  for i:=0 to knot-1 do
  for j:=0 to knot-1 do
  begin
    M:= (1/Factorial(knot-
1))*Permutation(knot-1,i);
    N:=0;
    for l:=j to knot-1 do
      N:=N+(Power(knot-l-1,i)*Power(-1,l-
j))*Permutation(knot,l-j));
    M:=M*N;
    Result[j+1,i+1]:=M;
  end;
end;

Function
TBSpline.SetB(o:integer;DTS: Variant): Variant;
var
  i,j:integer;
begin

Result:=varArrayCreate([1,knot,1,2],varDouble);
  for i:=0 to knot-1 do

```

```

begin
  Result[i+1,1]:=DTS[((i+o)mod(nSegment-
1))+1,1];
  Result[i+1,2]:=DTS[((i+o)mod(nSegment-
1))+1,2];
end;
end;

```

```

Function
TBSpline.SetOB(o:integer;DTS:Variant):Varia
nt;
var
  i,j:integer;
begin

```

```

  Result:=varArrayCreate([1,knot,1,2],varDoubl
e);

```

```

  for i:=0 to knot-1 do
    begin
      if(o+i<=nSegment)Then
        begin
          Result[i+1,1]:=DTS[o+i,1];
          Result[i+1,2]:=DTS[o+i,2];
        end
      else
        begin
          Result[i+1,1]:=DTS[o+i-nSegment,1];
          Result[i+1,2]:=DTS[o+i-nSegment,2];
        end;
      end;
    end;
  end;

```

```

end.

```



```

unit Main;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, NACA;

type
  TfmMain = class(TForm)
    Panel1: TPanel;
    btnQuit: TButton;
    btnParameter: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    btnDistrib: TButton;
    btnCavitation: TButton;
    btnOptimum: TButton;
    btnInfo: TButton;
    procedure btnParameterClick(Sender:
      TObject);
    procedure btnDistribClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnCavitationClick(Sender:
      TObject);
    procedure btnOptimumClick(Sender:
      TObject);
    procedure btnQuitClick(Sender: TObject);
    procedure btnInfoClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmMain: TfmMain;
  FNACA: TNACA;
implementation

uses Parameters, Distribution,
  CPGraph, InfoSoftware, Optimum;

{$R *.DFM}

procedure TfmMain.FormCreate(Sender:
  TObject);
begin
  FNACA:=Nil;
end;

procedure
  TfmMain.btnParameterClick(Sender:
  TObject);
var
  fmParameters: TfmParameters;
begin
  Try

```

```

    Application.CreateForm(TfmParameters, fmPar
      ameters);
    Finally
      fmParameters.ShowModal;
      fmParameters.Free;
    end;
  end;

  procedure TfmMain.btnDistribClick(Sender:
    TObject);
  var
    fmDistribution: TfmDistribution;
  begin
    If(FNACA<>Nil)Then
      Try

        Application.CreateForm(TfmDistribution, fmDi
          stribution);
        Finally
          fmDistribution.ShowModal;
          fmDistribution.Free;
        end
      else
        btnParameter.SetFocus;
    end;

    procedure
      TfmMain.btnCavitationClick(Sender:
      TObject);
    var
      fmCPGraph: TfmCPGraph;
    begin
      Try

        Application.CreateForm(TfmCPGraph, fmCPG
          raph);
        Finally
          fmCPGraph.ShowModal;
          fmCPGraph.Free;
        end;
      end;

      procedure TfmMain.btnOptimumClick(Sender:
        TObject);
      var
        fmOptimum: TfmOptimum;
      begin
        Try

          Application.CreateForm(TfmOptimum, fmOpti
            mum);
          Finally
            fmOptimum.ShowModal;
            fmOptimum.Free;
          end;
        end;

        procedure TfmMain.btnInfoClick(Sender:
          TObject);

```



```
var
fmInfo:TfmInfo;
begin
    Try
        Application.CreateForm(TfmInfo,fmInfo);
    Finally
        fmInfo.ShowModal;
        fmInfo.Free;
    end;
end;

procedure TfmMain.btnQuitClick(Sender:
TObject);
begin
    Close;
end;

end.
```

unit Parameters;

interface

uses

Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs,
ExtCtrls, StdCtrls, Grids, ComCtrls;

type

TfmParameters = class(TForm)
 Panel1: TPanel;
 Panel2: TPanel;
 Label1: TLabel;
 Label2: TLabel;
 ComboBox1: TComboBox;
 Panel3: TPanel;
 Label3: TLabel;
 Edit1: TEdit;
 btnPrint: TButton;
 btnExit: TButton;
 ComboBox2: TComboBox;
 Label9: TLabel;
 TabControl1: TTabControl;
 Image1: TImage;
 procedure FormCreate(Sender: TObject);
 procedure ComboBox1Change(Sender:
TObject);
 procedure ComboBox1KeyDown(Sender:
TObject; var Key: Word;
 Shift: TShiftState);
 procedure ComboBox2Change(Sender:
TObject);
 procedure ComboBox2KeyDown(Sender:
TObject; var Key: Word;
 Shift: TShiftState);
 procedure TabControl1Change(Sender:
TObject);
 procedure Edit1KeyDown(Sender: TObject;
var Key: Word;
 Shift: TShiftState);
 procedure Edit1Exit(Sender: TObject);
 procedure btnPrintClick(Sender: TObject);
 procedure btnExitClick(Sender: TObject);
 procedure FormClose(Sender: TObject; var
Action: TCloseAction);
private
 { Private declarations }
public
 { Public declarations }
end;

var

fmParameters: TfmParameters;
XShift, YShift: integer;
XScale, YScale: Double;

implementation

uses Main, NACA;

{ \$R *.DFM }

procedure TfmParameters.FormCreate(Sender:
TObject);
begin
 ComboBox1.ItemIndex:=9;
 XSHIFT:=10;
 YSHIFT:=Round(Image1.Height/2);
 XScale:=Image1.Width-20;
 YScale:=XScale;
 ComboBox1.OnChange(Self);
end;

procedure
TfmParameters.ComboBox1Change(Sender:
TObject);
var
 i: integer;
 STEP, TC, FC: Double;
begin
 If (ComboBox1.Text <> '') Then
 begin
 Val(Copy(ComboBox1.Text, 1, 1), FC, i);
 ComboBox2.Items.Clear;
 STEP:=0.5*FC/100;
 FC:=FC/100+2*STEP;
 for i:=0 to 2 do
 begin
 FC:=FC-STEP;
 ComboBox2.Items.Add(FloatToStrF(FC, ffFixed,
4, 4));
 end;
 TabControl1.Tabs.Clear;
 Val(Copy(ComboBox1.Text, 3, 2), TC, i);
 STEP:=TC/300;
 TC:=TC/100+3*STEP;
 for i:=0 to 4 do
 begin
 TC:=TC-STEP;
 TabControl1.Tabs.Add('T/C='+FloatToStrF(T
C, ffFixed, 4, 4));
 end;
 ComboBox2.ItemIndex:=1;
 ComboBox2.OnChange(Self);
 end;
end;

procedure
TfmParameters.ComboBox1KeyDown(Sender:
TObject; var Key: Word;
 Shift: TShiftState);
begin
 If (Key=13) Then
 ComboBox2.SetFocus;
end;

procedure
TfmParameters.ComboBox2Change(Sender:
TObject);
var


```

    i:integer;
begin
If((ComboBox1.Text<>")and(ComboBox2.Te
xt<>"))Then
TabControl1.TabIndex:=2;
Edit1.Text:='60';
Edit1.OnExit(Self);
end;

```

```

procedure
TfmParameters.ComboBox2KeyDown(Sender
: TObject; var Key: Word;
Shift: TShiftState);
begin
If(Key=13)Then
Edit1.SetFocus;
end;

```

```

procedure
TfmParameters.Edit1KeyDown(Sender:
TObject; var Key: Word;
Shift: TShiftState);
begin
If(Key=13)Then
btnPrint.SetFocus;
end;

```

```

procedure TfmParameters.Edit1Exit(Sender:
TObject);
begin
If((ComboBox1.Text<>")and(ComboBox2.Te
xt<>")and(Edit1.Text<>"))Then
begin
FNACA:=Nil;
FNACA:=TNACA.NACA(ComboBox1.Text,
StrToInt(Edit1.Text));
TabControl1.OnChange(Self);
end;
end;

```

```

procedure
TfmParameters.TabControl1Change(Sender:
TObject);
var
i,j:integer;
begin
Image1.Canvas.Pen.Color:=clBlack;
Image1.Canvas.Pen.Width:=1;
Image1.Canvas.Brush.Color:=clWhite;
Image1.Canvas.Rectangle(0,0,Image1.Width,I
mage1.Height);
i:=0;
While i<=Image1.Height do
begin
Image1.Canvas.MoveTo(0,i);
Image1.Canvas.LineTo(Image1.Width,i);
i:=i+2;
end;
i:=0;

```

```

While i<=Image1.Width do
begin
Image1.Canvas.MoveTo(i,0);
Image1.Canvas.LineTo(i,Image1.Height);
i:=i+2;
end;
Image1.Canvas.Pen.Width:=2;
Image1.Canvas.MoveTo(0,YSHIFT);
Image1.Canvas.LineTo(Image1.Width,YSHIF
T);
Image1.Canvas.Pen.Width:=3;
Image1.Canvas.Pen.Color:=clBlue;
If(FNACA<>Nil)Then
begin
Image1.Canvas.MoveTo(Round(FNACA.MLi
ne[ComboBox2.ItemIndex].PFoil[TabControl1
.TabIndex].XS[0]*XScale+XSHIFT),-
Round(FNACA.MLine[ComboBox2.ItemInde
x].PFoil[TabControl1.TabIndex].YS[0]*YScal
e)+YSHIFT);
for i:=1 to
High(FNACA.MLine[ComboBox2.ItemIndex]
.PFoil[TabControl1.TabIndex].XS)-1 do
Image1.Canvas.LineTo(Round(FNACA.MLin
e[ComboBox2.ItemIndex].PFoil[TabControl1.
TabIndex].XS[i]*XScale+XSHIFT),-
Round(FNACA.MLine[ComboBox2.ItemInde
x].PFoil[TabControl1.TabIndex].YS[i]*YScale
)+YSHIFT);
end;
end;

```

```

procedure
TfmParameters.btnPrintClick(Sender:
TObject);
begin
If(FNACA<>Nil)Then
begin
Panel3.Visible:=False;
Print;
Panel3.Visible:=True;
end
else
begin
ShowMessage('Fill in the Parameters First
!!!');
ComboBox1.SetFocus;
end;
end;

```

```

procedure
TfmParameters.btnExitClick(Sender:
TObject);
begin
Close;
end;

```

```

procedure TfmParameters.FormClose(Sender:
TObject;
var Action: TCloseAction);

```

```
begin
fmMain.btnDistrib.Enabled:=False;
fmMain.btnCavitation.Enabled:=False;
fmMain.btnOptimum.Enabled:=False;
If(FNACA<>Nil)Then
fmMain.btnDistrib.Enabled:=True;
end;

end.
```

```

unit Distribution;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Grids, ComCtrls, NACA;

type
  TfmDistribution = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    ComboBox1: TComboBox;
    Panel3: TPanel;
    Panel4: TPanel;
    Label3: TLabel;
    btnGraph: TButton;
    btnExit: TButton;
    Edit1: TEdit;
    TabControl1: TTabControl;
    StringGrid1: TStringGrid;
    Panel5: TPanel;
    Label4: TLabel;
    Label5: TLabel;
    ComboBox2: TComboBox;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender:
TObject);
    procedure ComboBox1KeyDown(Sender:
TObject; var Key: Word;
    Shift: TShiftState);
    procedure ComboBox2Change(Sender:
TObject);
    procedure ComboBox2KeyDown(Sender:
TObject; var Key: Word;
    Shift: TShiftState);
    procedure TabControl1Change(Sender:
TObject);
    procedure btnGraphClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var
Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmDistribution: TfmDistribution;
implementation

uses Main, PressGraph;

{$R *.DFM}

```

```

procedure
  TfmDistribution.FormCreate(Sender:
TObject);
var
  i,j:integer;
begin
  Edit1.Text:=FNACA.SERIES;
  ComboBox1.Items.Clear;
  For i:=0 to High(FNACA.MLine)do
    ComboBox1.Items.Add(FloatToStrF(FNACA.
MLine[i].FC,ffFixed,4,4));
  ComboBox2.Items.Clear;
  For i:=0 to High(FNACA.MLine[0].PFoil) do
    ComboBox2.Items.Add(FloatToStrF(FNACA.
MLine[0].PFoil[i].TC,ffFixed,4,4));
  ComboBox1.ItemIndex:=1;
  StringGrid1.RowCount:=2;
  StringGrid1.Cells[0,0]:='NO';
  StringGrid1.Cells[1,0]:='X';
  StringGrid1.Cells[2,0]:='Y';
  StringGrid1.Cells[3,0]:='THETA';
  StringGrid1.Cells[4,0]:='S';
  StringGrid1.Cells[5,0]:='GAMMA';
  StringGrid1.Cells[6,0]:='V';
  StringGrid1.Cells[7,0]:='CP';
  StringGrid1.Cells[8,0]:='CL';
  StringGrid1.Cells[0,1]:='';
  StringGrid1.Cells[1,1]:='';
  StringGrid1.Cells[2,1]:='';
  StringGrid1.Cells[3,1]:='';
  StringGrid1.Cells[4,1]:='';
  StringGrid1.Cells[5,1]:='';
  StringGrid1.Cells[6,1]:='';
  StringGrid1.Cells[7,1]:='';
  StringGrid1.Cells[8,1]:='';
  for i:=0 to High(FNACA.MLine) do
    for j:=0 to High(FNACA.MLine[i].PFoil)do
      FNACA.MLine[i].PFoil[j].SetPanel;
  ComboBox1.OnChange(Self);
end;

procedure
  TfmDistribution.ComboBox1Change(Sender:
TObject);
var
  i:integer;
begin
  If(ComboBox1.Text<>'')Then
  begin
    ComboBox2.ItemIndex:=2;
    ComboBox2.OnChange(Self);
  end;
end;

procedure
  TfmDistribution.ComboBox1KeyDown(Sende
r: TObject; var Key: Word;
  Shift: TShiftState);
begin
  If(Key=13)Then

```



```

ComboBox2.SetFocus;
end;

procedure
TfmDistribution.ComboBox2Change(Sender:
TObject);
begin
If(ComboBox2.Text<>"")Then
begin
TabControl1.TabIndex:=15;
TabControl1.OnChange(Self);
end;
end;

procedure
TfmDistribution.ComboBox2KeyDown(Sende
r: TObject; var Key: Word;
Shift: TShiftState);
begin
If(Key=13)Then
btnGraph.SetFocus;
end;

procedure
TfmDistribution.TabControl1Change(Sender:
TObject);
var
i:integer;
begin
StringGrid1.RowCount:=2;
StringGrid1.Cells[0,1]:="";
StringGrid1.Cells[1,1]:="";
StringGrid1.Cells[2,1]:="";
StringGrid1.Cells[3,1]:="";
StringGrid1.Cells[4,1]:="";
StringGrid1.Cells[5,1]:="";
StringGrid1.Cells[6,1]:="";
StringGrid1.Cells[7,1]:="";
If((ComboBox1.Text<>"")and(ComboBox1.Te
xt<>""))Then
begin
StringGrid1.RowCount:=FNACA.PANELCO
UNT+2;
for i:=1 to FNACA.PANELCOUNT+1 do
begin
StringGrid1.Cells[0,i]:=IntToStr(i);
StringGrid1.Cells[5,i]:=FloatToStrF(FNACA.
MLine[ComboBox1.ItemIndex].PFoil[Combo
Box1.ItemIndex].FPanels[TabControl1.TabInd
ex].GAMMA[i],ffFixed,6,6);
if(i>FNACA.PANELCOUNT+1)Then
begin
StringGrid1.Cells[1,i]:=FloatToStrF(FNACA.
MLine[ComboBox1.ItemIndex].PFoil[Combo
Box2.ItemIndex].FPanels[TabControl1.TabInd
ex].X[i],ffFixed,6,6);
StringGrid1.Cells[2,i]:=FloatToStrF(FNACA.
MLine[ComboBox1.ItemIndex].PFoil[Combo
Box2.ItemIndex].FPanels[TabControl1.TabInd
ex].Y[i],ffFixed,6,6);

```

```

StringGrid1.Cells[3,i]:=FloatToStrF(FNACA.
MLine[ComboBox1.ItemIndex].PFoil[Combo
Box2.ItemIndex].FPanels[TabControl1.TabInd
ex].THETA[i],ffFixed,6,6);
StringGrid1.Cells[4,i]:=FloatToStrF(FNACA.
MLine[ComboBox1.ItemIndex].PFoil[Combo
Box2.ItemIndex].FPanels[TabControl1.TabInd
ex].S[i],ffFixed,6,6);
StringGrid1.Cells[6,i]:=FloatToStrF(FNACA.
MLine[ComboBox1.ItemIndex].PFoil[Combo
Box2.ItemIndex].FPanels[TabControl1.TabInd
ex].VP[i],ffFixed,6,6);
StringGrid1.Cells[7,i]:=FloatToStrF(FNACA.
MLine[ComboBox1.ItemIndex].PFoil[Combo
Box2.ItemIndex].FPanels[TabControl1.TabInd
ex].CP[i],ffFixed,6,6);
end;
end;
Label4.Caption:='CPMIN =
'+FloatToStrF(FNACA.MLine[ComboBox1.It
emIndex].PFoil[ComboBox2.ItemIndex].FPan
els[TabControl1.TabIndex].CPMIN,ffFixed,6,
6);
end;
end;

```

```

procedure
TfmDistribution.btnGraphClick(Sender:
TObject);
var
fmGraph: TfmGraph;
begin
If(ComboBox1.Text<>"")Then
Try
Application.CreateForm(TfmGraph,fmGraph);
Finally
fmGraph.ShowModal;
fmGraph.Free;
end
else
ComboBox1.SetFocus;
end;
end;

```

```

procedure
TfmDistribution.btnExitClick(Sender:
TObject);
begin
Close;
end;
end;

```

```

procedure TfmDistribution.FormClose(Sender:
TObject;
var Action: TCloseAction);
begin
If(FNACA.Mline[0].PFoil[0].FPanels[0]<>Nil
)Then
begin
fmMain.btnCavitation.Enabled:=True;
fmMain.btnOptimum.Enabled:=True;

```



```
fmMain.btnCavitation.SetFocus;  
end  
else  
begin  
fmMain.btnCavitation.Enabled:=False;  
fmMain.btnOptimum.Enabled:=False;  
end;  
end;  
  
end.
```

```
unit PressGraph;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Grids, ComCtrls, BSPLine;
```

```
type
  TfmGraph = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    ComboBox1: TComboBox;
    Panel3: TPanel;
    Panel4: TPanel;
    Label3: TLabel;
    btnPrint: TButton;
    btnExit: TButton;
    Edit1: TEdit;
    TabControl1: TTabControl;
    Panel5: TPanel;
    Image1: TImage;
    Panel6: TPanel;
    Label4: TLabel;
    Label5: TLabel;
    ComboBox2: TComboBox;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender:
TObject);
    procedure ComboBox1KeyDown(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure ComboBox2Change(Sender:
TObject);
    procedure ComboBox2KeyDown(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure TabControl1Change(Sender:
TObject);
    procedure btnPrintClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
  fmGraph: TfmGraph;
  space1, space2, XShift, YShift: integer;
  XScale, YScale: Double;
  CPG: TBSPLine;
  CPP: Variant;
```

```
implementation
```

```
uses Main;
```

```
{ $R *.DFM }
```

```
procedure TfmGraph.FormCreate(Sender:
TObject);
var
  i: integer;
begin
  Edit1.Text:=FNACA.SERIES;
  CPP:=VarArrayCreate([1, FNACA.PANELCO
UNT, 1, 2], varDouble);
  space1:=Round((Image1.Width-10)/10);
  space2:=Round(Image1.Height/10);
  XShift:=0;
  YShift:=7*space2;
  XScale:=10*space1;
  YScale:=space2;
  ComboBox1.Items.Clear;
  For i:=0 to High(FNACA.MLine)do
    ComboBox1.Items.Add(FloatToStrF(FNACA.
MLine[i].FC, ffFixed, 4, 4));
  ComboBox2.Items.Clear;
  For i:=0 to High(FNACA.MLine[0].PFoil) do
    ComboBox2.Items.Add(FloatToStrF(FNACA.
MLine[0].PFoil[i].TC, ffFixed, 4, 4));
  ComboBox1.ItemIndex:=1;
  ComboBox1.OnChange(Self);
end;
```

```
procedure
  TfmGraph.ComboBox1Change(Sender:
TObject);
begin
  ComboBox2.ItemIndex:=2;
  ComboBox2.OnChange(Self);
end;
```

```
procedure
  TfmGraph.ComboBox1KeyDown(Sender:
TObject; var Key: Word;
  Shift: TShiftState);
begin
  If(Key=13)Then
    ComboBox2.SetFocus;
  end;
```

```
procedure
  TfmGraph.ComboBox2Change(Sender:
TObject);
begin
  If((ComboBox1.Text<>'')and(ComboBox2.Te
xt<>''))Then
  begin
    TabControl1.TabIndex:=15;
    TabControl1.OnChange(Self);
  end;
end;
```



```

procedure
TfmGraph.ComboBox2KeyDown(Sender:
TObject; var Key: Word;
Shift: TShiftState);
begin
If(Key=13)Then
btnPrint.SetFocus;
end;

procedure
TfmGraph.TabControl1Change(Sender:
TObject);
var
i,j:integer;
begin
If((ComboBox1.Text<>'')and(ComboBox2.Te
xt<>''))Then
begin
for i:=1 to FNACA.PANELCOUNT do
begin
CPP[i,1]:=FNACA.MLine[ComboBox1.ItemI
ndex].PFoil[ComboBox2.ItemIndex].XB[i];
CPP[i,2]:=FNACA.MLine[ComboBox1.ItemI
ndex].PFoil[ComboBox2.ItemIndex].FPanels[
TabControl1.TabIndex].CP[i];
end;
CPG:=TBSPLine.OBSPLine(CPP,2);
Image1.Canvas.Pen.Color:=clBlack;
Image1.Canvas.Pen.Width:=1;
Image1.Canvas.Brush.Color:=ClWhite;
Image1.Canvas.Rectangle(0,0,Image1.Width,I
mage1.Height);
Image1.Canvas.Pen.Color:=clBlack;
Image1.Canvas.Pen.Width:=1;
for i:=1 to 9 do
begin
Image1.Canvas.MoveTo(i*space1,0);
Image1.Canvas.LineTo(i*space1,Image1.Heig
ht);
end;
for i:=1 to 9 do
begin
Image1.Canvas.MoveTo(0,i*space2);
Image1.Canvas.LineTo(Image1.Width,i*space
2);
end;
Image1.Canvas.Pen.Width:=2;
Image1.Canvas.MoveTo(0,0);
Image1.Canvas.LineTo(0,Image1.Height);
Image1.Canvas.MoveTo(0,7*space2);
Image1.Canvas.LineTo(Image1.Width,7*space
2);
Image1.Canvas.TextOut(1,1,'CP');
Image1.Canvas.TextOut(Image1.Width-
25,7*space2-15,'X/C');
for i:=1 to 9 do
Image1.Canvas.TextOut(i*space1+1,7*space2
+1,'0.'+IntToStr(i));
j:=1;
for i:=1 to 3 do

```

```

begin
Image1.Canvas.TextOut(1,j*space2-1,'-
'+IntToStr(4-i));
j:=j+2;
end;
Image1.Canvas.TextOut(1,7*space2+1,'0');
Image1.Canvas.TextOut(1,9*space2-1,'1');
Image1.Canvas.Pen.Width:=3;
Image1.Canvas.Pen.Color:=clBlue;
Image1.Canvas.MoveTo(CPG.BSP[1,1]*10*S
pace1,CPG.BSP[1,2]*Space1+7*space2);
for i:=2 to CPG.nData do
Image1.Canvas.LineTo(CPG.BSP[i,1]*10*Spa
ce1,CPG.BSP[i,2]*Space1+7*space2);
Label4.Caption:='CPMIN =
'+FloatToStrF(FNACA.MLine[ComboBox1.It
emIndex].PFoil[ComboBox2.ItemIndex].FPan
els[TabControl1.TabIndex].CPMIN,ffFixed,6,
6);
end;
end;

procedure TfmGraph.btnPrintClick(Sender:
TObject);
begin
Panel3.Visible:=False;
Print;
Panel3.Visible:=True;
end;

procedure TfmGraph.btnExitClick(Sender:
TObject);
begin
Close;
end;

end.

```

```

unit CPGraph;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Grids, ComCtrls, Math;

type
  TfmCPGraph = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Panel3: TPanel;
    Panel4: TPanel;
    btnPrint: TButton;
    btnExit: TButton;
    Edit1: TEdit;
    Image1: TImage;
    Label9: TLabel;
    ComboBox1: TComboBox;
    Option: TRadioGroup;
    Label3: TLabel;
    Edit2: TEdit;
    btnTabel: TButton;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender:
TObject);
    procedure btnPrintClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure ComboBox1KeyDown(Sender:
TObject; var Key: Word;
    Shift: TShiftState);
    procedure OptionClick(Sender: TObject);
    procedure OptionExit(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
    procedure btnTabelClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmCPGraph: TfmCPGraph;
  space1, space2, XShift, YShift: integer;
  marked, XScale, YScale: Double;
implementation

uses Main, TabelForm;

{$R *.DFM}

```

```

procedure TfmCPGraph.FormCreate(Sender:
TObject);
var
  i: integer;
begin
  marked:=0;
  Edit1.Text:=FNACA.SERIES;
  ComboBox1.Items.Clear;
  If (Option.ItemIndex=0) Then
  begin
    for i:=0 to High(FNACA.MLine) do
      ComboBox1.Items.Add(FloatToStr(FNACA.
MLine[i].FC));
    ComboBox1.ItemIndex:=1;
  end
  else
  begin
    for i:=0 to High(FNACA.MLine[0].PFoil) do
      ComboBox1.Items.Add(FloatToStr(FNACA.
MLine[0].PFoil[i].TC));
    ComboBox1.ItemIndex:=2;
  end;
  space1:=Round((Image1.Width-10)/6);
  space2:=Round(Image1.Height/17);
  XShift:=0;
  YShift:=9*space2;
  XScale:=2*space1;
  YScale:=space2;
  ComboBox1.OnChange(Self);
end;

procedure
TfmCPGraph.ComboBox1Change(Sender:
TObject);
var
  i,j,k: integer;
  p,q: Double;
begin
  Image1.Canvas.Pen.Color:=clBlack;
  Image1.Canvas.Pen.Width:=1;
  Image1.Canvas.Brush.Color:=clWhite;
  Image1.Canvas.Rectangle(0,0,Image1.Width,I
mage1.Height);
  for i:=1 to 5 do
  begin
    Image1.Canvas.MoveTo(i*space1,0);
    Image1.Canvas.LineTo(i*space1,Image1.Heig
ht);
  end;
  for i:=1 to 16 do
  begin
    Image1.Canvas.MoveTo(0,i*space2);
    Image1.Canvas.LineTo(Image1.Width,i*space
2);
  end;
  Image1.Canvas.Pen.Width:=2;
  Image1.Canvas.MoveTo(0,YShift);
  Image1.Canvas.LineTo(Image1.Width,YShift);
  Image1.Canvas.MoveTo(0,0);
  Image1.Canvas.LineTo(0,Image1.Height);

```



```

Image1.Canvas.TextOut(1,1,'Ang of
Attack(Deg)');
Image1.Canvas.TextOut(Image1.Width-
40,YShift-15,'-CPMIN');
p:=0;
for i:=1 to 6 do
begin
p:=p+0.5;
Image1.Canvas.TextOut(i*space1+1,YShift+1,
FloatToStr(p));
end;
for i:=1 to 8 do
Image1.Canvas.TextOut(1,i*space2+1,IntToSt
r(9-i));
for i:=1 to 7 do
Image1.Canvas.TextOut(1,(9+i)*space2+1,Int
ToStr(-i));
Image1.Canvas.TextOut(1,YShift+1,'0');
Image1.Canvas.Pen.Width:=2;
Image1.Canvas.Pen.Color:=clBlue;
If(Option.ItemIndex=0)Then
begin
Label4.Visible:=True;
Label5.Visible:=True;
Label6.Visible:=True;
Label7.Visible:=True;
Label8.Visible:=True;
Label4.Caption:='Blue: T/C =
'+FloatToStr(FNACA.MLine[ComboBox1.It
emIndex].PFoil[0].TC);
Label5.Caption:='Red: T/C =
'+FloatToStr(FNACA.MLine[ComboBox1.It
emIndex].PFoil[1].TC);
Label6.Caption:='Yellow: T/C =
'+FloatToStr(FNACA.MLine[ComboBox1.It
emIndex].PFoil[2].TC);
Label7.Caption:='Green: T/C =
'+FloatToStr(FNACA.MLine[ComboBox1.It
emIndex].PFoil[3].TC);
Label8.Caption:='Maroon: T/C =
'+FloatToStr(FNACA.MLine[ComboBox1.It
emIndex].PFoil[4].TC);
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[0].FPanels[0].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[0].FPanels[0].Attack);
Image1.Canvas.MoveTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[ComboBox1.ItemIndex]
.PFoil[0].FPanels)do
begin
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[0].FPanels[j].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[0].FPanels[j].Attack);
Image1.Canvas.LineTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);

```

```

end;
Image1.Canvas.Pen.Color:=clRed;
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[1].FPanels[0].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[1].FPanels[0].Attack);
Image1.Canvas.MoveTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[ComboBox1.ItemIndex]
.PFoil[1].FPanels)do
begin
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[1].FPanels[j].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[1].FPanels[j].Attack);
Image1.Canvas.LineTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
end;
Image1.Canvas.Pen.Color:=clYellow;
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[2].FPanels[0].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[2].FPanels[0].Attack);
Image1.Canvas.MoveTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[ComboBox1.ItemIndex]
.PFoil[2].FPanels)do
begin
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[2].FPanels[j].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[2].FPanels[j].Attack);
Image1.Canvas.LineTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
end;
Image1.Canvas.Pen.Color:=clGreen;
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[3].FPanels[0].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[3].FPanels[0].Attack);
Image1.Canvas.MoveTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[ComboBox1.ItemIndex]
.PFoil[3].FPanels)do
begin
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[3].FPanels[j].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[3].FPanels[j].Attack);
Image1.Canvas.LineTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);

```



```

end;
Image1.Canvas.Pen.Color:=clMaroon;
p:=FNACA.MLine[ComboBox1.ItemIndex].PFoil[4].FPanels[0].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.ItemIndex].PFoil[4].FPanels[0].Attack);
Image1.Canvas.MoveTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[ComboBox1.ItemIndex].PFoil[4].FPanels)do
begin
p:=FNACA.MLine[ComboBox1.ItemIndex].PFoil[4].FPanels[j].CPMIN;
q:=RadToDeg(FNACA.MLine[ComboBox1.ItemIndex].PFoil[4].FPanels[j].Attack);
Image1.Canvas.LineTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
end;
end
else
begin
Label4.Visible:=True;
Label5.Visible:=True;
Label6.Visible:=True;
Label7.Visible:=False;
Label8.Visible:=False;
Label4.Caption:='Blue: F/C =
'+FloatToStr(FNACA.MLine[0].FC);
Label5.Caption:='Red: F/C =
'+FloatToStr(FNACA.MLine[1].FC);
Label6.Caption:='Yellow: F/C =
'+FloatToStr(FNACA.MLine[2].FC);
Image1.Canvas.Pen.Color:=clBlue;
p:=FNACA.MLine[0].PFoil[ComboBox1.ItemIndex].FPanels[0].CPMIN;
q:=RadToDeg(FNACA.MLine[0].PFoil[ComboBox1.ItemIndex].FPanels[0].Attack);
Image1.Canvas.MoveTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[0].PFoil[ComboBox1.ItemIndex].FPanels)do
begin
p:=FNACA.MLine[0].PFoil[ComboBox1.ItemIndex].FPanels[j].CPMIN;
q:=RadToDeg(FNACA.MLine[0].PFoil[ComboBox1.ItemIndex].FPanels[j].Attack);
Image1.Canvas.LineTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
end;
end;
Image1.Canvas.Pen.Color:=clRed;
p:=FNACA.MLine[1].PFoil[ComboBox1.ItemIndex].FPanels[0].CPMIN;
q:=RadToDeg(FNACA.MLine[1].PFoil[ComboBox1.ItemIndex].FPanels[0].Attack);

```

```

Image1.Canvas.MoveTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[1].PFoil[ComboBox1.ItemIndex].FPanels)do
begin
p:=FNACA.MLine[1].PFoil[ComboBox1.ItemIndex].FPanels[j].CPMIN;
q:=RadToDeg(FNACA.MLine[1].PFoil[ComboBox1.ItemIndex].FPanels[j].Attack);
Image1.Canvas.LineTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
end;
end;
Image1.Canvas.Pen.Color:=clYellow;
p:=FNACA.MLine[2].PFoil[ComboBox1.ItemIndex].FPanels[0].CPMIN;
q:=RadToDeg(FNACA.MLine[2].PFoil[ComboBox1.ItemIndex].FPanels[0].Attack);
Image1.Canvas.MoveTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[2].PFoil[ComboBox1.ItemIndex].FPanels)do
begin
p:=FNACA.MLine[2].PFoil[ComboBox1.ItemIndex].FPanels[j].CPMIN;
q:=RadToDeg(FNACA.MLine[2].PFoil[ComboBox1.ItemIndex].FPanels[j].Attack);
Image1.Canvas.LineTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
end;
end;
If(marked>0)Then
begin
Image1.Canvas.Pen.Width:=2;
Image1.Canvas.Pen.Color:=clPurple;
Image1.Canvas.MoveTo(Round(marked*XScale)+XShift,0);
Image1.Canvas.LineTo(Round(marked*XScale)+XShift,Image1.Height);
end;
end;

procedure
TfmCPGraph.ComboBox1KeyDown(Sender:
TObject; var Key: Word;
Shift: TShiftState);
begin
If(Key=13)Then
btnPrint.SetFocus;
end;

procedure TfmCPGraph.btnPrintClick(Sender:
TObject);
begin
Panel3.Visible:=False;

```

```

Print;
Panel3.Visible:=True;
end;

procedure TfmCPGraph.btnExitClick(Sender:
TObject);
begin
    Close;
end;

procedure TfmCPGraph.OptionClick(Sender:
TObject);
var
    i:integer;
begin
    If (Option.ItemIndex=0)Then
        Label9.Caption:='F/C : '
    else
        Label9.Caption:='T/C : ';
        ComboBox1.Items.Clear;
        If (Option.ItemIndex=0)Then
            begin
                for i:=0 to High(FNACA.MLine)do
                    ComboBox1.Items.Add(FloatToStr(FNACA.
                    MLine[i].FC));
                ComboBox1.ItemIndex:=1;
            end
        else
            begin
                for i:=0 to High(FNACA.MLine[0].PFoil)do
                    ComboBox1.Items.Add(FloatToStr(FNACA.
                    MLine[0].PFoil[i].TC));
                ComboBox1.ItemIndex:=2;
            end;
            ComboBox1.OnChange(Self);
        end;

    procedure TfmCPGraph.OptionExit(Sender:
    TObject);
    begin
        ComboBox1.SetFocus;
    end;

    procedure TfmCPGraph.Edit2Change(Sender:
    TObject);
    begin
        If (Edit2.Text<>'')Then
            begin
                marked:=StrToFloat(Edit2.Text);
                ComboBox1.OnChange(Self);
            end;
        end;

    procedure
    TfmCPGraph.btnTabelClick(Sender: TObject);
    var
        fmTabel:TfmTabel;
    begin

```

```

        Try
            Application.CreateForm(TfmTabel,fmTabel);
            Finally
                fmTabel.ShowModal;
                fmTabel.Free;
            end;
        end;

    end.

```



```

unit CPGPanel;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Grids, ComCtrls, Math;

type
  TfmCPGPanel = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Panel3: TPanel;
    Panel4: TPanel;
    btnPrint: TButton;
    btnExit: TButton;
    Edit1: TEdit;
    Label9: TLabel;
    ComboBox1: TComboBox;
    TabControl1: TTabControl;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender:
TObject);
    procedure btnPrintClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure ComboBox1KeyDown(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure TabControl1Change(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmCPGPanel: TfmCPGPanel;
  space1, space2, XShift, YShift: integer;
  XScale, YScale: Double;
implementation

uses Main;

{$R *.DFM}

procedure TfmCPGPanel.FormCreate(Sender:
TObject);
var
  i: integer;
begin
  Edit1.Text := FNACA.SERIES;
  ComboBox1.Items.Clear;
  for i:=0 to High(FNACA.MLine) do
    ComboBox1.Items.Add(FloatToStr(FNACA.
MLine[i].FC));

```

```

    ComboBox1.ItemIndex:=1;
    space1:=Round((Image1.Width-10)/6);
    space2:=Round(Image1.Height/17);
    XShift:=0;
    YShift:=9*space2;
    XScale:=2*space1;
    YScale:=space2;
    ComboBox1.OnChange(Self);
  end;

  procedure
    TfmCPGPanel.ComboBox1Change(Sender:
TObject);
  var
    i: integer;
  begin
    If(ComboBox1.Text<>'')Then
      begin
        TabControl1.Tabs.Clear;
        for i:=0 to
          High(FNACA.MLine[ComboBox1.ItemIndex]
.PFoil)do
          TabControl1.Tabs.Add(FloatToStr(FNACA.M
Line[ComboBox1.ItemIndex].PFoil[i].TC));
        TabControl1.TabIndex:=2;
        TabControl1.OnChange(Self);
      end;
    end;

  procedure
    TfmCPGPanel.TabControl1Change(Sender:
TObject);
  var
    i,j,k: integer;
    p,q: Double;
  begin
    Image1.Canvas.Pen.Color:=clBlack;
    Image1.Canvas.Pen.Width:=1;
    Image1.Canvas.Brush.Color:=clWhite;
    Image1.Canvas.Rectangle(0,0,Image1.Width,I
mage1.Height);
    for i:=1 to 5 do
      begin
        Image1.Canvas.MoveTo(i*space1,0);
        Image1.Canvas.LineTo(i*space1,Image1.Heig
ht);
      end;
    for i:=1 to 16 do
      begin
        Image1.Canvas.MoveTo(0,i*space2);
        Image1.Canvas.LineTo(Image1.Width,i*space
2);
      end;
    Image1.Canvas.Pen.Width:=2;
    Image1.Canvas.MoveTo(0,YShift);
    Image1.Canvas.LineTo(Image1.Width,YShift);
    Image1.Canvas.MoveTo(0,0);
    Image1.Canvas.LineTo(0,Image1.Height);
    Image1.Canvas.TextOut(1,1,'Ang of
Attack(Deg)');

```



```

Image1.Canvas.TextOut(Image1.Width-
40,YShift-15,'CP');
p:=0;
for i:=1 to 6 do
begin
p:=p+0.5;
Image1.Canvas.TextOut(i*space1+1,YShift+1,
FloatToStr(p));
end;
for i:=1 to 8 do
Image1.Canvas.TextOut(1,i*space2+1,IntToSt
r(9-i));
for i:=1 to 7 do
Image1.Canvas.TextOut(1,(9+i)*space2+1,Int
ToStr(-i));
Image1.Canvas.TextOut(1,YShift+1,'0');
Image1.Canvas.Pen.Width:=1;
k:=clBlue-10;
for i:=1 to FNACA.PANELCOUNT do
begin
k:=k+50;
Image1.Canvas.Pen.Color:=k;
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[TabControl1.TabIndex].FPanels[0].CP[i];
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[TabControl1.TabIndex].FPane
ls[0].Attack);
Image1.Canvas.MoveTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[ComboBox1.ItemIndex]
.PFoil[ComboBox1.ItemIndex].FPanels)do
begin
p:=FNACA.MLine[ComboBox1.ItemIndex].P
Foil[TabControl1.TabIndex].FPanels[j].CP[i];
q:=RadToDeg(FNACA.MLine[ComboBox1.It
emIndex].PFoil[TabControl1.TabIndex].FPane
ls[j].Attack);
Image1.Canvas.LineTo(-
Round(p*XScale)+XShift,-
Round(q*YScale)+YShift);
end;
end;
end;

procedure
TfmCPGPanel.ComboBox1_KeyDown(Sender:
TObject; var Key: Word;
Shift: TShiftState);
begin
If(Key=13)Then
btnPrint.SetFocus;
end;

procedure
TfmCPGPanel.btnPrintClick(Sender:
TObject);
begin
Panel3.Visible:=False;

```

```

Print;
Panel3.Visible:=True;
end;

procedure TfmCPGPanel.btnExitClick(Sender:
TObject);
begin
Close;
end;

end.

```



```

unit CLGraph;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Grids, ComCtrls;

type
  TfmCLGraph = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Panel3: TPanel;
    Panel4: TPanel;
    btnPrint: TButton;
    btnExit: TButton;
    Edit1: TEdit;
    Label3: TLabel;
    ComboBox1: TComboBox;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender:
TObject);
    procedure btnPrintClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmCLGraph: TfmCLGraph;
  space1, space2, XShift, YShift: integer;
  XScale, YScale: Double;
implementation

uses Main;

{$R *.DFM}

procedure TfmCLGraph.FormCreate(Sender:
TObject);
var
  i: integer;
begin
  Edit1.Text:=FNACA.SERIES;
  space1:=Round((Image1.Width-10)/8);
  space2:=Round(Image1.Height/13);
  XShift:=0;
  YShift:=10*space2;
  XScale:=2.5*space1;
  YScale:=5*space2;
  ComboBox1.Items.Clear;
  for i:=0 to High(FNACA.Mline) do
    ComboBox1.Items.Add(FloatToStr(FNACA.
Mline[i].FC));

```

```

    ComboBox1.ItemIndex:=1;
    ComboBox1.OnChange(Self);
  end;

procedure
TfmCLGraph.ComboBox1Change(Sender:
TObject);
var
  i,j,k: integer;
  a,b,p: Double;
begin
  If(ComboBox1.Text<>'')Then
  begin
    Image1.Canvas.Pen.Color:=clBlack;
    Image1.Canvas.Pen.Width:=1;
    Image1.Canvas.Brush.Color:=clWhite;
    Image1.Canvas.Rectangle(0,0,Image1.Width,I
mage1.Height);
    Image1.Canvas.Pen.Color:=clBlack;
    Image1.Canvas.Pen.Width:=1;
    for i:=1 to 7 do
    begin
      Image1.Canvas.MoveTo(i*space1,0);
      Image1.Canvas.LineTo(i*space1,Image1.Heig
ht);
    end;
    for i:=1 to 12 do
    begin
      Image1.Canvas.MoveTo(0,i*space2);
      Image1.Canvas.LineTo(Image1.Width,i*space
2);
    end;
    Image1.Canvas.Pen.Width:=2;
    Image1.Canvas.MoveTo(0,YShift);
    Image1.Canvas.LineTo(Image1.Width,YShift);
    Image1.Canvas.MoveTo(0,0);
    Image1.Canvas.LineTo(0,Image1.Height);
    Image1.Canvas.TextOut(5,1,'CL');
    Image1.Canvas.TextOut(Image1.Width-
20,YShift-15,'-CP');
    p:=0;
    for i:=1 to 7 do
    begin
      p:=p+0.4;
      Image1.Canvas.TextOut(i*space1+1,YShift+1,
FloatToStr(p));
    end;
    p:=2;
    for i:=1 to 9 do
    begin
      p:=p-0.2;
      Image1.Canvas.TextOut(1,i*space2+1,FloatTo
Str(p));
    end;
    Image1.Canvas.TextOut(1,YShift+1,'0');
    p:=0;
    for i:=1 to 2 do
    begin
      p:=p-0.2;

```

```

Image1.Canvas.TextOut(1,(10+i)*space2+1,FloatToStr(p));
end;
Image1.Canvas.Pen.Width:=1;
k:=clBlue-50;
for k:=0 to
High(FNACA.MLine[ComboBox1.ItemIndex].PFoil) do
for i:=1 to FNACA.PANELCOUNT do
begin
a:=FNACA.MLine[ComboBox1.ItemIndex].PFoil[k].FPanels[0].CP[i];
b:=FNACA.MLine[ComboBox1.ItemIndex].PFoil[k].FPanels[0].CL[i];
Image1.Canvas.MoveTo(-
Round(a*XScale)+XShift,-
Round(b*YScale)+YShift);
for j:=1 to
High(FNACA.MLine[ComboBox1.ItemIndex].PFoil[i].FPanels) do
begin
a:=FNACA.MLine[ComboBox1.ItemIndex].PFoil[k].FPanels[j].CP[i];
b:=FNACA.MLine[ComboBox1.ItemIndex].PFoil[k].FPanels[j].CL[i];
Image1.Canvas.LineTo(-
Round(a*XScale)+XShift,-
Round(b*YScale)+YShift);
end;
end;
end;
end;

procedure TfmCLGraph.btnPrintClick(Sender:
TObject);
begin
Panel3.Visible:=False;
Print;
Panel3.Visible:=True;
end;

procedure TfmCLGraph.btnExitClick(Sender:
TObject);
begin
Close;
end;

end.

```